

**uc3m** | Universidad **Carlos III** de Madrid

Grado Universitario Electronica Industrial y  
Automatica  
2016/2017

*Trabajo Fin de Grado*

# **ON-BOARD VISUAL TRACKING WITH UNMANNED AERIAL VEHICLES**

---

Abraham García Amez

Tutor

Abdulla Hussein Al Kaff

Lugar y fecha de presentación prevista



*[Incluir en el caso de interés en su publicación en el archivo abierto]*

Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**

# Resumen

En este documento se describen los aspectos más importantes del Sistema de Seguimiento de objetos. El objetivo principal de este trabajo consiste en la construcción de un sistema de detección y seguimiento de objetos, por medio de un sistema óptico de visión monocular, con el que se realiza la detección de un objeto fijo en tiempo real utilizándose para su -reconocimiento un patrón pintado en el objetivo.

Para el correcto uso de la cámara, será necesario realizar su calibración de esta para obtener sus parámetros intrínsecos y extrínsecos, la distancia focal.

Por otra parte, los datos calculados serán enviados a la parte de control del UAV que se encargará de tratar esta información para actualizar sobre el objetivo fijo según sea necesario.

Con este sistema también se podría conseguir el seguimiento de objetos en movimiento, mediante selección del objetivo o marcando con el patrón a detectar.

# Abstract

This document describes the most important aspect of the Tracking Objects System. The main aim of this task consists in building a detection and tracking object system, through a optical system whit monocular vision, which is used to detect a fixed target in real time, using a pattern painted in the target in order to recognise it.

For the correct use of the camera, it calibration is necessary to get the parameters intrinsic and extrinsics, the focal length.

On the other hand, calculated data will be sent to the control system of UAV that it will use this information to act about the fixed target as necessary.

Whit this system,tracking on movement also could be got, through the selection of the target or making the pattern to detect.

# Índice

<b>1. <u>Introducción</u></b>	<b>6</b>
1.1. Motivación . . . . .	7
1.2. Problemas que nos podemos encontrar . . . . .	9
1.3. Solución a ofrecer para los problemas . . . . .	10
1.4. Objetivos . . . . .	11
1.5. Resumen de lo que se vera en el documento . . . . .	11
<b>2. <u>Estado del Arte</u></b>	<b>13</b>
2.1. Introducción . . . . .	14
2.2. UAVs . . . . .	14
2.2.1. Historia UAV . . . . .	14
2.2.2. Clasificación . . . . .	17
2.2.3. Trabajos con cámaras en los UAVs . . . . .	20
2.2.4. Otros sistemas de localización . . . . .	24
2.3. Obtención y descripción de los keypoints . . . . .	28
2.3.1. Método SURF . . . . .	30
2.3.2. Método SIFT . . . . .	34
2.3.3. Método FAST . . . . .	39
2.3.4. Método AGAST . . . . .	42
<b>3. <u>Algoritmo</u></b>	<b>44</b>
3.1. Aspectos generales . . . . .	45
3.1.1. Objetivos . . . . .	45
3.1.2. Datos de entrada . . . . .	47
3.1.3. Datos de Salida . . . . .	48
3.2. Algoritmo Teórico . . . . .	48
3.3. Desarrollo del algoritmo . . . . .	49
3.3.1. Obtención y discriminación de los keypoints . . . . .	49
3.3.2. Homografía y búsqueda de objeto . . . . .	53
3.3.3. Coordenadas y altura respecto al objeto . . . . .	55
<b>4. <u>Experimentos y resultados</u></b>	<b>59</b>
4.1. Introducción . . . . .	60
4.2. Plataforma . . . . .	60
4.2.1. Hardware . . . . .	60
4.2.2. Software . . . . .	60
4.3. Experimentos . . . . .	61
<b>5. <u>Conclusiones y futuros trabajos</u></b>	<b>67</b>
5.1. Introducción . . . . .	68
5.2. Contribuciones del proyecto . . . . .	68
5.3. Conclusión . . . . .	68
5.4. Trabajos futuros . . . . .	69

<b>6. Apéndice</b>	<b>71</b>
6.1. Apéndice A . . . . .	72
6.2. Apéndice B . . . . .	73

## Índice de figuras

1.	A la izq: Despliegue de estaciones. A la der: Estacion frontal. . . . .	9
2.	Cronología UAVs . . . . .	15
3.	MALE UAS Pedrator . . . . .	16
4.	Yamaha R50 . . . . .	16
5.	Modelo Python . . . . .	17
6.	Clasificación UAV según su despegue . . . . .	18
7.	UAV multi-rotor y UAV helicóptero . . . . .	19
8.	UAV Ala delta y UAV parapente . . . . .	19
9.	UAV aeroplano . . . . .	20
10.	Sistema de Nuno Pessanha Santos . . . . .	23
11.	Funcionamiento basico ultrasonido . . . . .	26
12.	Rango de distancias de aplicación de los sistemas de posicionamiento	26
13.	Estructura y tamaño de los filtros utilizados en SURF a diferentes escalas. . . . .	31
14.	Función de Harr para el método SURF . . . . .	33
15.	Diferencia Gaussianas en la escala-espacio . . . . .	36
16.	Ciruclo Braseham de circulo 3 . . . . .	40
17.	Diagrama reducido del código . . . . .	46
18.	Tipos de escalas de imagen. A la izq escala en gris. A la der escala en color . . . . .	50
19.	Matriz intrinseca con $z=0$ . . . . .	53
20.	Trigonometría . . . . .	56
21.	Distancia focal . . . . .	57
22.	Experimento imagen 1 con surf . . . . .	62
23.	Experimento imagen 1 con sift . . . . .	62
24.	Experimento imagen 2 con método SURf . . . . .	63
25.	Experimento imagen 2 con método SIFT . . . . .	64
26.	Experimento imagen 3 con método AGAST . . . . .	65
27.	Experimento imagen 3 con método SURF . . . . .	65
28.	Experimento imagen 4 con método AGAST . . . . .	66

# 1. Introducción



## 1.1. Motivación

El uso de cámaras para ayudar a las personas son cada vez más comunes. Estas aplicaciones se dan tanto en seguridad, control, mantenimiento, etc dentro de estas se usan en el transporte, viviendas, ciudades, etc. Por ejemplo, en el ámbito de transporte en los vehículos es muy común que lleven cámaras para avisar al conductor de que se va a salir de su carril o que se esta quedando dormido, etc. En el ámbito de ciudades se llegan a usar para el control de gente en un espacio deseado, por ejemplo, para saber cuanta gente entra y sale, seguimiento de alguien por reconocimiento facial.

En este documento como UAV hemos usado un dron de cuatro motores el más sencillo, que se le ha incorporado una cámara monocular para el visionado del terreno. Los drones son máquinas que podemos controlar mediante un sistema remoto o pueden navegar de forma autónoma, son capaces de portar herramientas específicas para ciertos trabajos específicos, también puedes acceder a sitios donde el ser humano correría peligro o incluso no serían capaces de acceder. Y si se colisionan producen menos costes que si hubiera un accidente con un helicóptero o avionetas. Por estas ventajas que tiene el uso de drones hay muchos programadores que están centrando sus estudios en esta rama, creando aplicaciones y usos para estos aparatos.

Unos de los campos que más futuro tiene dentro de las aplicaciones con drones es el procesador de imágenes en tiempo real. Esta tecnología nos permite ciertas ventajas, llevar una cámara a rincones inalcanzables para el ser humano con costes económicos y con mínimo personal.

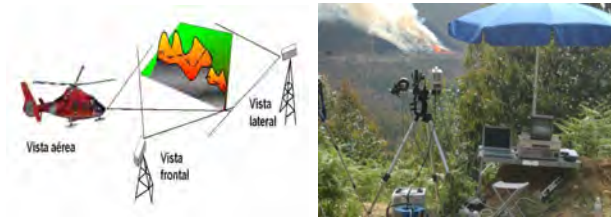
Las cámaras también se van incorporando más en los distintos ámbitos, por ejemplo, en el ámbito deportivo para ayudar a saber que ha pasado en el momento de dictar una sanción al jugador, también se van incorporando con programas específicos según el uso que se le vaya a dar, por ejemplo, control de un acceso de vehículos mediante sus matriculas en un parking privado.

Para conseguir este proyecto nos haremos uso de algoritmos de imágenes en movimiento para la extracción y detección de características de imágenes.

Tenemos dos ejemplo de proyecto civil que emplean más de dos cámaras para la visión, el primer caso de un sistema para la medición automática y monitorización de fuegos [J. R. Martínez-de Dios, A. Ollero, L. Merino, y B. C. Arrue, 2006] mediante cámaras visuales y de infrarrojos desarrollado por el Grupo de Robótica, Visión y Control de la Universidad de Sevilla. Este sistema es capaz de medir automáticamente y en tiempo real parámetros del frente de llamas tales como la posición, la velocidad y altura de llamas empleando cámaras visuales y de infrarrojos con diferentes vistas en localizaciones fijas o en medios terrestres o aéreos.

Y el segundo ejemplo, emplean dos cámaras para la obtención de la información del entorno, de forma que con ambas se pueden obtener, mediante el procesamiento de las imágenes, las coordenadas 3D buscadas. Un ejemplo de estos sistemas de visión estereoscópica es un robot de gran precisión diseñado para la desactivación de bombas [Hengnian Qi, Wei Wang, Liangzhong Jiang, Luqiao Fan, 2007]. Tras la localización de la bomba, el manipulador será controlado para llegar a ella una vez conocidas sus coordenadas. El sistema del robot está formado por dos partes, el subsistema de visión estereoscópica y el subsistema de control de movimiento.

También hay otros sistemas emplea una cámara combinada con otro sensor, se puede mencionar un sistema de medida de visión a distancia para vehículos submarinos no tripulados [Muljowidodo K., Mochammad. A. Rasyid., 18Identificación óptica de la posición y orientación de un Vehículo Aéreo no Tripulado Sapto Adi N., and Agus Budiyo] que emplea una cámara y un puntero láser. Este sistema predice la distancia horizontal entre el vehículo y la pared o la distancia vertical entre el vehículo y la superficie por debajo de él. Por medio del procesamiento de la imagen, detecta el láser y calcula la distancia basada en la posición del láser en la imagen.



**Figura 1:** A la izq: Despliegue de estaciones. A la der: Estacion frontal.

## 1.2. Problemas que nos podemos encontrar

Tanto con los UAV y las cámaras nos encontramos problemas de hardware. En los UAV sobre todo el problema esta en la duración de las baterías que nos encontramos con duraciones aproximadamente menores a la hora de uso y si no se está atento del tiempo de vuelo y si no dispone el UAV de un sistema de prevención de aterrizaje automático cuando la batería llega a un porcentaje critico, el UAV caerá sin control, provocando daños al UAV al caer contra el suelo a gran velocidad.

En las cámaras el problema que nos podemos encontrar es la calidad de la cámara, por ejemplo a que resolución, frame por segundo a la que puede grabar la cámara, también el ruido, las distorsiones, falsos colores, etc, que producen los sensores que incorporan las cámaras.

Para nuestro sistema de aterrizaje automático el problema que nos podemos encontrar son varios, como hemos hablado antes los UAV y las cámaras tienen ciertos inconvenientes por lo que al mezclarlo estos problemas se suman y se pueden complicar. Por ejemplo el aterrizaje automático necesita de un patrón donde se quiera aterrizar el UAV, ya sea encima de un vehículo, navío o incluso en una plataforma terrestre. Si este patrón no esta bien dibujado, esta desgastado e incluso no haya buena iluminación el sistema de aterrizaje no detectara bien el patrón y si el patrón no se detecta correctamente, puede que el sistema no calculé bien la altura y esto pueda provocar dos problemas, el primero es que el sistema crea que está más arriba y en realidad este más cerca del suelo y el UAV se estrelle o al revés, que el sistema crea que esta en el suelo pero en realidad el UAV se encuentre a una altura elevada.

Para el aterrizaje automático también nos podemos encontrar con problemas en el control de la velocidad en el descenso, este fallo también viene producido por lo descrito en el párrafo anterior. Si el sistema calcula que el UAV está a una cierta altitud elevada el descenso puede ser más rápido pero si en realidad se encuentra más bajo, el UAV al ir deprisa podrá ocasionar ciertos daños e incluso su propia destrucción.

### 1.3. Solución a ofrecer para los problemas

En los UAV la solución que podemos aportar es poner baterías de más amperios horas para tener mayor duración y también incorporar un sistema para prevenir que el UAV se quede sin batería y caiga en vacío. Este sistema sería que el control del UAV detectara el nivel de carga de las baterías y cuando estas lleguen a un determinado nivel, así el sistema aterrizara el UAV en la localización que se encuentre o si le permite retornar al punto de partida o ir a un punto elegido por el usuario.

En las cámaras no podemos modificar el hardware por eso todas las soluciones que se pueden hacer son de tipo software. Cuando se está haciendo el programa para la aplicación que queremos, se meten ciertos comandos que mejoran o modifican la calidad de la imagen para mejorar las características que nos interesen para que la aplicación funcione correctamente. Estos comandos lo que hacen es mejorar bordes, quitar el ruido que produce los sensores, quitar la distorsión que se puede producir si la cámara no enfoca correctamente, etc.

La solución que ofrecemos para los problemas que hemos descrito cuando el sistema está montado en conjunto, es nuestro sistema de aterrizaje automático, ya que prevendremos y tendremos en cuenta todos esos problemas que hemos descrito anteriormente y así el aterrizaje automático sea el mejor posible y el que menos inconvenientes ponga al UAV y así no le dañe.

Para el control de la velocidad de los motores en su aterrizaje nos apoyaremos en un proyecto realizado por un compañero.

## 1.4. Objetivos

El objetivo de este proyecto es la adquisición de los puntos característicos que compone la imagen patrón para más tarde poder encontrarlos en la imagen a tiempo real. Para conseguirlo, se debe atender a unas etapas, que pasaremos a explicar:

- Obtención de la matriz de calibración de la cámara: Se obtiene la matriz de la cámara para poder obtener ciertos valores específicos para el calculo de la altura.
- Obtención de los puntos característicos: Se obtendrán los puntos característicos de cada imagen obtenida por la cámara. Se expondrán varias formas de extracción de las características de dichos puntos y la elección de los más adecuados.
- Comparación de los puntos y eliminación de los puntos no deseados: Cuando se tengan los puntos característicos de las imágenes, se procederá a la comparación entre las imágenes. A través de este método se obtiene los puntos que pertenecen a las dos imágenes y el resto al ser puntos no deseados se eliminaran.
- Calculo de posición y orientación: Una vez detectado el objeto se procederá a la obtención de la posición y orientación del patrón respecto al centro de la imagen.

## 1.5. Resumen de lo que se vera en el documento

La memoria estará estructurada en cinco bloques. Se van a proceder a exponer un breve resumen de cada uno a continuación:

- Capitulo 1 Introducción. En este capítulo se introduce las motivaciones por la que se elabora el proyecto y los objetivos a los que se procederá a conseguir.
- Capitulo 2 Estado del Arte. En este capítulo se muestra la base teórica que es necesaria para la comprensión y la elaboración del proyecto. Se definirán los métodos de obtención de las puntos característicos, SIFT, SURF, AKAZE Y ORB.
- Capitulo 3 Algoritmo. Este capitulo se define el algoritmo que se ha usado para la obtención de los puntos característicos reales y su uso. Para ello se expondrán

cada una de las partes en las que está dividido, como es la extracción de los keypoints y descriptors, la comparación entre las dos imágenes y calculo de posición y orientación.

-Capítulo 4 Experimentos y Resultados. Este capítulo se introducen las pruebas prácticas que se han obtenido de la ejecución del algoritmo. Se efectúa la justificación de la elección del método elegido para la extracción de los keypoints. Y también se comentará los componentes utilizados en el proyecto.

-Capítulo 5 Conclusiones y trabajos futuros. Se expondrán las conclusiones obtenidas del proyecto y también las líneas de investigación futuras por la que se pueda avanzar el proyecto.

## 2. Estado del Arte

## 2.1. Introducción

En este capítulo vamos a hablar sobre la historia de los UAVs, la forma con la que se pueden clasificar, su composición. También hablaremos de las aplicaciones que se le puede dar a la cámara y pondremos algunos ejemplos.

## 2.2. UAVs

La palabra UAV viene de las sigas Unmanned Aerial Vehicle, en castellano se traduce como Vehículo Aéreo No Tripulado, en siglas VANT, la agencia de defensa del ejercito de Estados unidos[2] lo define como:

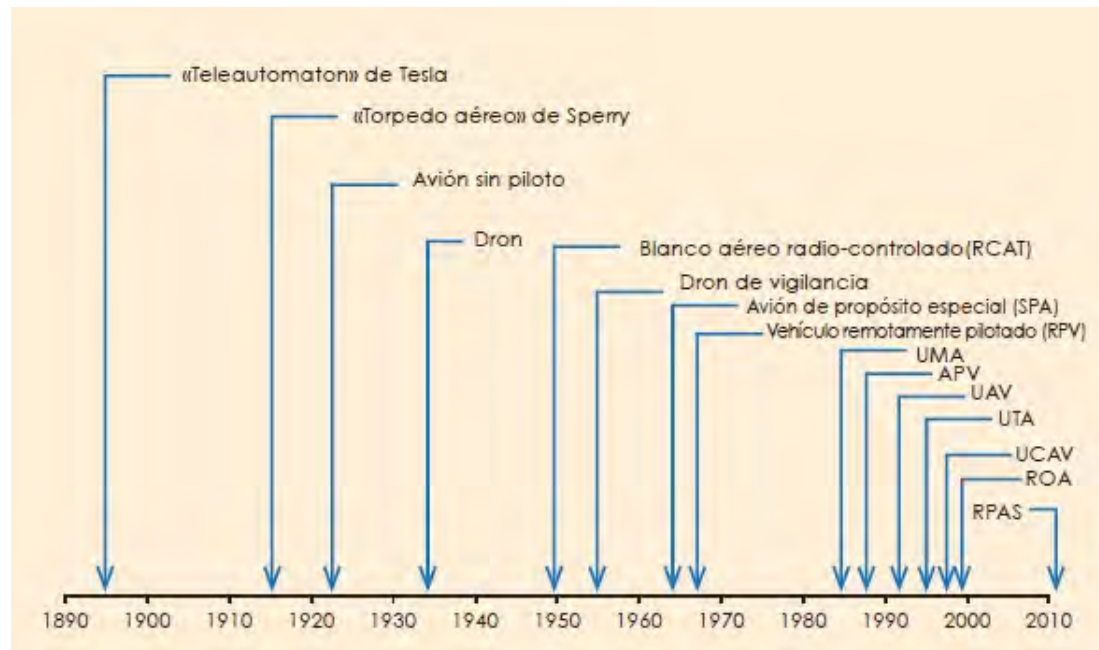
Un vehículo aéreo que no lleva a bordo a un operador humano, utiliza las fuerzas aerodinámicas para generar la sustentación, puede volar de forma autónoma o remota, que puede ser fungible o recuperable, y que puede transportar una carga de pago letal o no.

Por lo que esta definición excluye a los misiles, planeadores(no llevan planta propulsora) o los globos y dirigibles de la clasificación de UAVs.

### 2.2.1. Historia UAV

A partir de a Segunda Guerra Mundial se ha intentado conseguir un arma que consiguiera generar daños sin poner siendo capaz de no arriesgar la vida humana. Por eso los ejércitos de las dos grandes superpotencias de la época (Unión Soviética y Estados Unidos) han investigado en nuevos aviones que puedan ser pilotadas desde tierra, con la consecuencia de no poner en peligro a ningún soldado. [12]





**Figura 2:** Cronología UAVs

En las décadas posteriores a la Segunda Guerra Mundial se experimentaban para conseguir la obtención del UAV. Los primeros UAV que se obtuvieron, fueron avionetas controladas por radiofrecuencia. El inconveniente de este tipo de control era su fiabilidad y su corto alcance de trabajo, por lo que solo se usó este método para experimentación.

Cuando se desarrollaron los aviones a reacción se vuelve a intensificar la investigación del poder manejar los aviones de forma remota. Durante la década de los sesenta se introducen dispositivos de visión, usando se para espionaje y para reconocimiento del terreno.

Pero estos tipos de UAV también presentaban inconvenientes: era muy difícil de hacerlos aterrizar sin que se ocasionara daños en la aeronave. Otros problemas que surgía en esa época eran las limitaciones tecnológicas.

Gracias a la mejora que hubo en la tecnología de la década de los años ochenta se pudo intensificar el uso de los UAVs. Se pudieron desarrollar vuelos pre-programados y se empezó a usar los helicópteros. También se fueron introduciendo más sensores que permitían hacer la navegación y el control desde tierra más fácil.

El estudio de naves auto-tripuladas siguió en aumento por el ejercito de los Estados Unidos. Introdujeron mejoras en las naves, por ejemplo, el sistema de GPS, uso de sistemas digitales de control de vuelo, y también el vuelo a mayor altura para conseguir ser no descubierto.

En la época de los noventa se desarrolla el primer UAV militar funcional, el cual se denomina MALE UAS Pedrator. Este UAV es de gran envergadura, con propulsión a reacción y se puede controlar desde tierra a tiempo real.



**Figura 3:** MALE UAS Pedrator

También aparece las primeras aeronaves de despegue y aterrizaje en vertical.



**Figura 4:** Yamaha R50

Paralelamente al desarrollo de los UAV militares, se empezó a ver que utilidad se le puede a dar a los UAV en la vida civil. Pero se ve retenido por problemas éticos sobre su uso indebido.

Por otro lado, se potencia el uso de las naves de despegue y aterrizaje vertical debido a que tienen un mejor uso y un menor índice de accidentes.

A partir del año 2010, van apareciendo nuevos UAV, los UAV para uso comercial. Estos UAV hacen un gran impacto sobre el público por su tamaño reducido y el manejo que se consigue, se les denomina como drones. Al ser controlados fácilmente mediante un radio control, se van introduciendo en el uso de actividades deportivas u obtención de imágenes.

Pero aún no se queda hay su utilidad, ya que estos dispositivos tienen muchas posibilidades de uso, ya sea para uso personal o uso profesional.

En conclusión, el mundo de los drones dispone de una gran evolución y por esto los investigadores deciden invertir en esta tecnología.



**Figura 5:** Modelo Python

### **2.2.2. Clasificación**

Los UAV se pueden clasificar con distintos criterios, por su la carga máxima, la forma de despegue, la altitud a la que es capaz de alcanzar o el nivel de autonomía

alcanzable[5].

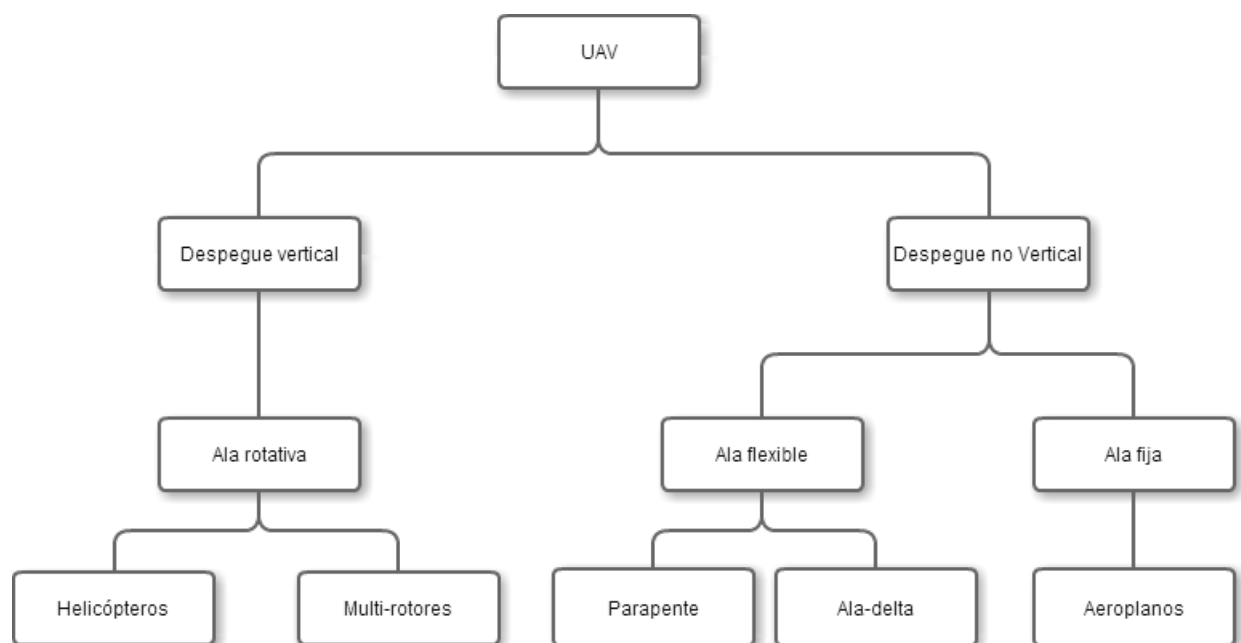
Vamos a diferenciar la diferencia que hay entre el término UAS y el término UAV:

UAS (Unmanned Aerial System): abarca tanto la nave, sensores, sistema de control, que puede ser autónomo o ser controlado desde tierra.

UAV (Unmanned Aerial Vehicle): abarca la nave, sensores y define la propulsión, procesado, estructura y sensores.

Como no se puede distinguir a primera vista los términos, en el proyecto vamos a usar el término UAV para simplificar.

En el documento clasificaremos los UAV según su forma de despegue, ya que nos dará una mejor aproximación al aspecto.

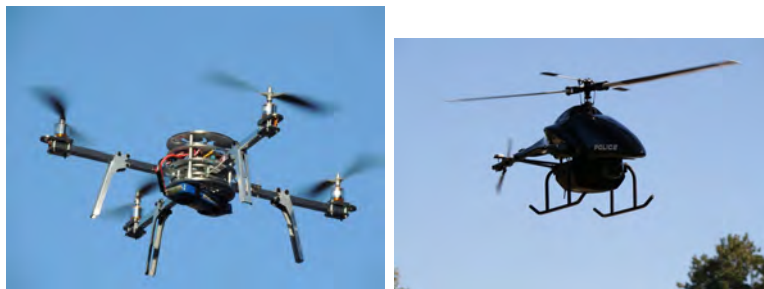


**Figura 6:** Clasificación UAV según su despegue

Dentro de la clasificación de UAV de despegue vertical se puede diferenciar por su sistema de estabilidad, los helicópteros consiguen la estabilidad a través del rotor de cola y los multi-rotors consiguen la estabilidad a través de la alternancia de la aceleración de sus motores.

Para los UAV de despegue horizontal podemos clasificarlo por el tipo de ala. Los aviones de ala fija, utilizan un propulsor y un timonel de cola consiguen así su estabilidad. Los UAV que están basados en el ala de un ala delta consiguen la estabilidad variando la superficie de contacto con el aire. Los UAV de ala flexible se desplazan gracias a un propulsor y consiguen la estabilidad gracias a un sistema basado en ala delta.

Los UAV que más se usan para el uso comercial son de despegue vertical (pala rotativa), ya que la acción de despejar y aterrizar es menos peligroso y más sencillo.



**Figura 7:** UAV multi-rotor y UAV helicóptero



**Figura 8:** UAV Ala delta y UAV parapente



**Figura 9:** UAV aeroplano

### 2.2.3. Trabajos con cámaras en los UAVs

La ventaja que se consigue al usar un UAV es la posibilidad de volar en casi cualquier zona con ventajas que no se consiguen con naves pilotadas:

- Un coste reducido de mantenimiento.
- El riesgo que sufren los trabajadores se minimiza al no haber pilotos.
- Se puede acceder a las zonas más difíciles que naves más grandes.

Para que el usuario pueda ver lo que contempla el UAV, se le añade una cámara. Por lo que la cámara se convierte en uno de los elementos más importantes a desarrollar, la integración de la adquisición de imágenes y su análisis.

Se va a proceder a explicar algún ejemplo civil para demostrar que el tratamiento de las imágenes tiene mucha importancia.

Vamos a empezar la aplicación de UAV en el mantenimiento de tendidos eléctricos. Como el uso de tendidos eléctricos aéreos son muy comunes para el transporte de la electricidad. Para poder mantener estos postes hasta hace unos años el sistema de mantenimiento era costoso y arriesgado ya que se tenía que usar helicópteros. Se ha empezado a usar los UAV para este tipo de trabajo, con lo que se consigue el descenso del coste de mantenimiento y sobre todo la disminución del riesgo que corren los trabajadores.

Los UAVs en el mantenimiento de los postes se centran sobre todo en los siguientes campos:

- Tendido de cables: Cuando se van a instalar nuevos postes de tendido eléctrico

y se tienen que tirar los cables de alta tensión entre los dos postes es necesario tirar un cable guía, por el cual después va a ir el cable principal. El cable guía se suele llevar de un poste a otro con ayuda del UAV, con lo que se consigue salvar accidentes geográficos.

-Transporte de cargas: Para ayudar a los operarios en la reparación de los postes, si necesitan herramientas o incluso piezas de sustitución, en vez de bajar y subir con el UAV se les facilita sin tener que moverse del poste.

-Inspección aérea de las líneas: Para esta acción se suele usar helicópteros, pero con los UAV se consigue ahorrar en costes de mantenimiento y sobre todo minimizar el riesgo de dañar el tendido.

-Inspección intensiva de líneas: Para este trabajo el trabajador se debe subir al poste lo que con lleva un riesgo de caída, por lo que hoy en día se están empleando los UAV. Estos aportar seguridad al trabajador y la inspección se realiza en un tiempo menor.

Se puede consultar el artículo[7] que aparece en la bibliografía.

En la agricultura también se han empezado a usar los UAVs, al usar este sistema se ha conseguido aumentar la producción respecto a años atrás. Con los UAVs se consigue cierta precisión, por lo que se consigue aprovechar mejor los recursos. En la agricultura se puede distinguir en cuatro etapas:

-Monitorización: Es la detección y el mapeo de las variables que se desean en cada momento.

-Toma de decisiones: Es la elaboración de un plan para la resolución de los problemas.

-Actuación: La ejecución del plan que se ha propuesto.

-Comprobación: Evaluación de si ha sido eficaz el plan, tanto medioambiental como económico, para poder ser aplicados el año que viene.

En la parte de monitorización es cuando entra en funcionamiento el UAV. Antes de que se usara los UAVs, se usaba imágenes por satélites, pero este sistema tenía un inconveniente y es que no se podía usar cuando estaba nublado y sobre todo



las imágenes son de baja resolución. El uso del UAV se aplica a:

- Detención de necesidad de riesgo para cada zona.
- Detención de plagas.
- Detención de malas hierbas en las zonas.

Se puede consultar el artículo[9][11] que aparece en la bibliografía.

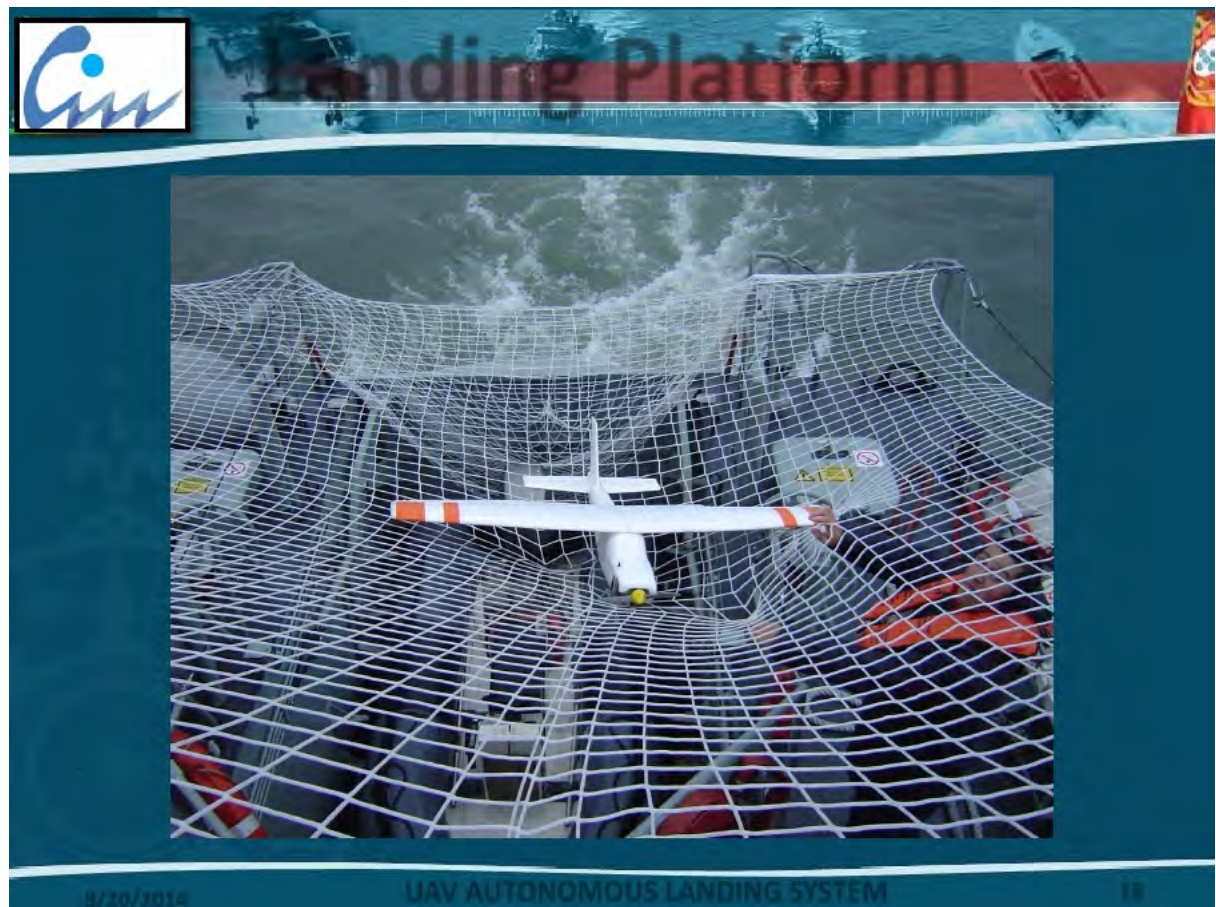
También hay que tener en cuenta el número de cámaras empleadas, depende de las necesidades del sistema, algunos sistemas cuentan con una única cámara y los sistemas que emplean dos cámaras, llevando a cabo lo que se conoce como visión estereoscópica, y otros con más de dos. Todos estos sistemas, además, pueden complementarse con el uso de algún sensor de distinta naturaleza.

A continuación se van a mencionar un ejemplo de un proyecto que tiene que detectar y seguir un objeto y que emplean la visión.

En el caso de un robot jugador de Ping-Pong [J. J. Rodrigo, L. Acosta, J.A. Méndez, S. Torres], por ejemplo, su sistema de visión está formado por una única cámara y éste se basa en la detección de la pelota y de la sombra que la misma proyecta sobre la mesa. Las tareas del sistema de visión son las de procesar la imagen y estimar la trayectoria que seguirá la pelota en el espacio. Su cuerpo, además, consta de dos articulaciones que permiten llevar a cabo las tareas del sistema de control, interceptar la bola y orientarla en el espacio, y golpearla según la estrategia de juego.

Ahora se van a mostrar un ejemplo de aterrizajes automatizados de otros usuarios. El primer ejemplo que vamos a poner es un sistema del aterrizaje de un UAV sobre un navío en movimiento [Nuno Pessanha Santos]. Este sistema utiliza la visión y mezcla varios métodos de detección de objetos para conseguir que el UAV pueda aterrizar sin ningún inconveniente.





**Figura 10:** Sistema de Nuno Pessanha Santos

Otro ejemplo de aterrizaje automático es de la universidad Chemnitz de Alemania [Sven Lange, Niko Sünderhauf, Peter Protzel], este ejemplo es de un quadrotor que tiene que aterrizar sobre un patrón. Este sistema es parecido a nuestro sistema por lo que podríamos compararlo con nuestro sistema. Este sistema es distinto al anterior descrito ya que la cámara está en posición para ir siempre paralela al suelo como nuestro sistema, por lo que nos deshacemos de un inconveniente que tener la cámara algo inclinada. Al tener la cámara paralela al suelo el cálculo de la posición respecto al patrón se hace mucho más simple.

Como se ha podido observar, son numerosos los sistemas que emplean la visión para el reconocimiento del medio.

Nuestro sistema se enmarcara dentro del primer grupo, es decir, el sistema se encargara de la localización de un objeto fijo. Como se ha explicado anteriormente, la idea inicial se baso en un cuatrimotor que, tras detectar el patrón elegido podrá calcular su posición, altura y la orientación respecto al patrón para descender en el.

#### **2.2.4. Otros sistemas de localización**

A continuación se realiza un repaso por los distintos sistemas sensoriales [A. Gardel Vicente, 2004], existentes en la actualidad, que permiten la localización de un objeto en el entorno, analizando sus características.

Los robots cuentan en su arquitectura básica con sensores. Estos sensores, a pesar de ser de diferentes tipos y naturaleza, son complementarios, de hecho, la fusión de información proveniente de distintos sistemas sensoriales ha sufrido un considerable auge.

Una posible clasificación de los sensores está basada en la forma de controlar el robot, de manera interna o externa:

- Sensores internos: controlan accionamientos de la estructura mecánica del robot y permiten implementar técnicas de posicionamiento relativo como los potenciómetros o los sensores odométricos e inerciales.

- Sensores externos: detectan la posición del robot respecto al entorno. Estos sensores pueden ser de contacto (antenas, bumpers, dedo robot, array sensor, etc.) o de no contacto como los ultrasonidos, la visión, los infrarrojos o los sensores radio.

Otra posible clasificación de los sensores se basa en el posicionamiento llevado a cabo, es decir, relativo o absoluto:

- Sensores relativos: miden la posición del robot a partir de su posición ante-

rior, por lo que es necesario conocer su posición inicial, pero presentan el problema de que, con el tiempo, acumulan errores de cálculo, haciendo necesaria una nueva localización externa. Estos sensores son los odométricos y los inerciales, de los que algunos ejemplos son los potenciómetros, encoders, giróscopos y acelerómetros.

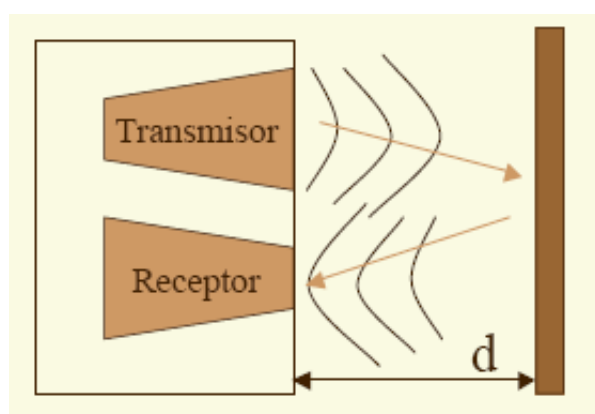
- Sensores absolutos: miden la posición del robot respecto un sistema de referencia externo, independientemente de su posición anterior. Estas técnicas son usadas por sensores externos. Los sensores explicados a continuación son sensores absolutos. En tareas de posicionamiento a corta distancia donde es necesaria una localización muy precisa, los sensores empleados son los mecánicos y los magnéticos. Los sensores mecánicos no tienen problemas de interferencias y sus errores están sujetos a las incertidumbres de las medidas de las piezas y holguras en los movimientos mecánicos. Mientras que los sensores magnéticos son más baratos que los mecánicos y no tienen que estar en contacto con el objeto pero sí muy próximos a él.

Dentro de los posibles sensores ópticos, la luz infrarroja es utilizada frecuentemente para tener menos interferencias con otras fuentes luminosas, pasando inadvertida para los humanos. Los sensores ópticos tienen una elevada precisión y rápida respuesta aunque están limitados por el alcance de la fuente de luz. El precio de estos sensores es más elevado que el de otros.

Dentro de los sistemas de visión artificial se puede hacer uso de marcas artificiales y/o marcas naturales. La ventaja de las artificiales con respecto a las naturales es que son más sencillas de reconocer en el entorno porque su apariencia no depende del entorno, hecho que hace a éstas más robustas aunque, por el hecho de existir ya en el entorno, las naturales no hay que incorporarlas.

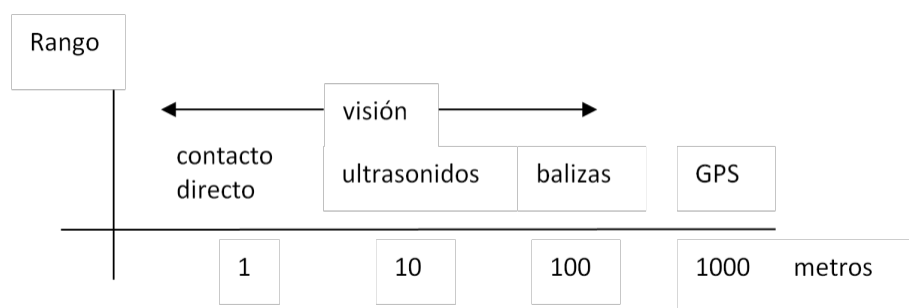
Los dispositivos de ultrasonidos [Diego Pérez de Diego, 2006] utilizan frecuencias por encima de los 20KHz, imperceptibles por el oído humano. Son baratos, compactos y ligeros, pero su precisión no es muy elevada y depende de las condiciones del medio en el que se encuentren.

El funcionamiento básico de los ultrasonidos se muestra en la siguiente figura, donde se tiene un receptor que emite vibraciones periódicas en el tiempo y en el espacio a una velocidad de transmisión de 340m/seg. Estas vibraciones rebotan sobre un determinado objeto y la reflexión de esas vibraciones es detectada por un receptor de ultrasonidos. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia a la que se encuentra el objeto que ha producido la reflexión de la onda sonora.



**Figura 11:** Funcionamiento basico ultrasonido

Los sistemas radio, GSM, etc. se utilizan para la localización y posicionamiento robusto de vehículos y móviles, pero tienen una precisión baja y son susceptibles de interferencias por otras ondas de radio.



**Figura 12:** Rango de distancias de aplicación de los sistemas de posicionamiento

Como se puede ver en la figura anterior, según para qué distancias se trate, son

mejores algunos sistemas u otros.

A distancias cortas, los sistemas más útiles son los de contacto directo (sensores mecánicos y magnéticos) ya que realizan los cálculos con gran precisión.

Para distancias alrededor de los 10m los dispositivos más adecuados son los ultrasonidos. Este sensor espera un tiempo hasta que las vibraciones emitidas desaparezcan y esté preparado para recibir el eco producido por el objeto, por lo que existe una distancia mínima a partir de la cual el sensor mide con precisión. A distancias más pequeñas, el sensor, por tanto, no es preciso.

Sin embargo, las balizas permiten posicionar un objeto a 100m de distancia mediante la comparación de señales enviadas por las fuentes emisoras. A distancias de 1000m, el sistema más adecuado es el GPS, Global Positioning System, o Sistema de posicionamiento y navegación basado en satélites.

En el rango de distancias de 0 a 100m, se puede emplear la visión para el posicionamiento de un objeto. En el apartado anterior “Trabajos relacionados” se pueden comprobar las distintas utilidades de los sistemas de visión y las distintas precisiones conseguidas en ellos. Por ejemplo, el sistema desarrollado para la desactivación de bombas [Hengnian Qi, Wei Wang, Liangzhong Jiang, Luqiao Fan, 2007], necesita unos cálculos muy precisos y a muy pequeñas distancias, frente al de monitorización de incendios [J. R. Martínez-de Dios, A. Ollero, L. Merino, y B. C. Arrue, 2006] en el que la precisión es importante, pero no se consideran distancias tan cortas y por tanto, el error en los cálculos es admisible.

Tras este análisis de los dispositivos sensoriales existentes, cabe destacar que nuestro sistema lleva a cabo el posicionamiento de manera absoluta, puesto que no es necesario en un momento dado conocer la posición anterior del cuatrimotor.

El sistema emplea la visión como medio de comunicación con el entorno puesto que, por medio de las capturas de la cámara, se obtiene la información necesaria

para el cálculo de la posición y de la orientación respecto al patrón.

También hay que indicar que las marcas empleadas para la visión son artificiales, pudiendo emplear también marcas naturales, siempre y cuando sean bien distinguibles en el entorno.

En nuestro caso, con la cámara de que disponemos y las pruebas realizadas con el sistema, la distancia de aplicación para el posicionamiento y la precisión dependen directamente del tamaño del objeto de referencia.

Si el objeto de referencia es pequeño, la precisión será mayor cuanto más cerca esté el objeto de la cámara; o lo que es lo mismo, a mayor distancia, más posibilidades hay de que las cámaras no vean el objeto o se confunda con el entorno y, por lo tanto, menos fiabilidad habrá en el cálculo de las medidas.

Sin embargo, si el objeto de referencia es más grande, se puede calcular la posición a mayor distancia, siempre y cuando el objeto esté en el campo de visión de la cámara.

### **2.3. Obtención y descripción de los keypoints**

Los keypoints (puntos de interés) es uno de los aspectos más fundamental en la computación de imágenes. EL objetivo es encontrar los puntos de la imagen que nos defina el contenido de la imagen. En esta sección vamos a explicar el concepto de keypoint, explicando sus características y el tipo de clasificación de las mismas. Los descriptors es la forma de almacenar la información que nos ofrece los keypoints.

Se van a explicar algunos de los métodos que se pueden usar para la extracción de los puntos y los descriptors y por último, se va a realizar una comparación de los métodos que se van a describir y se elegirá el que mejor se adapte a nuestro

objetivo.

Definir los keypoints tiene una complejidad, ya que según sea la aplicación en la que se vaya a utilizar, la definición puede ser más o menos compleja. Un keypoint de una imagen contiene cualidades o características que se destacan de una imagen. Las características pueden variar según el proceso que se vaya a hacer.

Para poder hacer la detección de las características se necesita una operación de la imagen de bajo nivel (low-level), es decir, son operaciones que se hacen en primer lugar. El algoritmo de detección aplica una combinación de filtros que suavizan la imagen, con lo que se consigue eliminar el ruido que producen los sensores de las cámaras e información innecesaria. El siguiente paso es la búsqueda en cada píxel las posibles características y esta información se guarda en los descriptores, que son un vector.

Estos datos contienen información de los píxeles vecinos, así se puede definir un cambio de tendencia en la imagen, con lo que conlleva información valiosa en el tratamiento de imágenes. Los keypoints más destacables suelen estar dentro de los siguientes grupos:

- Esquinas: Grupo de píxeles que están agrupados y dan un brusco cambio de dirección. Esta clasificación agrupa las esquinas tradicionales, como las esquinas que se aprecian una curvatura.

- Bordes: Cuando un píxel o un conjunto de ellos sirve de frontera entre dos zonas de píxeles con información diferente. Estos puntos contienen un alto gradiente en la imagen, con lo que se puede identificar el brusco cambio de tendencia entre píxeles.

- Arrugas: poder distinguir bordes en zonas con un límite muy arrugado es difícil de conseguir. Por lo que estos puntos se llaman arrugas y se pueden clasificar como bordes.

-Blobs: Gotas en castellano, se hace referencia a las zonas que no han sido suavizados, por lo que no se puede detectar como esquina ni como keypoint.

Como se ha comentado anteriormente en los vectores denominados descriptor se guarda la información. En cada posición del vector se guarda la información que proporciona el keypoint y la posición que se corresponde en la imagen.

Como cada algoritmo de extracción de las características de la imagen tiene sus ventajas e inconvenientes, vamos a exponer una muestra de algunos de los métodos de extracción de los keypoints, método SURF, método SIFT, método FAST, método AGAST.

### 2.3.1. Método SURF

SURF (Speeded Up Robust Features) hace mención a un algoritmo de localización y almacenamiento de los keypoints. Fue desarrollado por Herbert Bay[6] y su primera publicación fue en 2006 en la Conferencia Europea de Visión por Computador. Este método de obtención de keypoints consta en varias etapas:

- Identificación de keypoints.
- Determinación de la orientación.
- Construcción del descriptor.

## IDENTIFICACION KEYPOINTS

Para la identificación de los keypoints, el método SURF hace una aproximación de la matriz Hessiana para determinar la escala y posición, obteniendo así una mayor velocidad de cálculo y también consiguiendo una alta precisión.

Para cualquier punto  $p(x, y)$  en la imagen  $I$ , la matriz Hessiana a escala  $\sigma$  se



describe como:

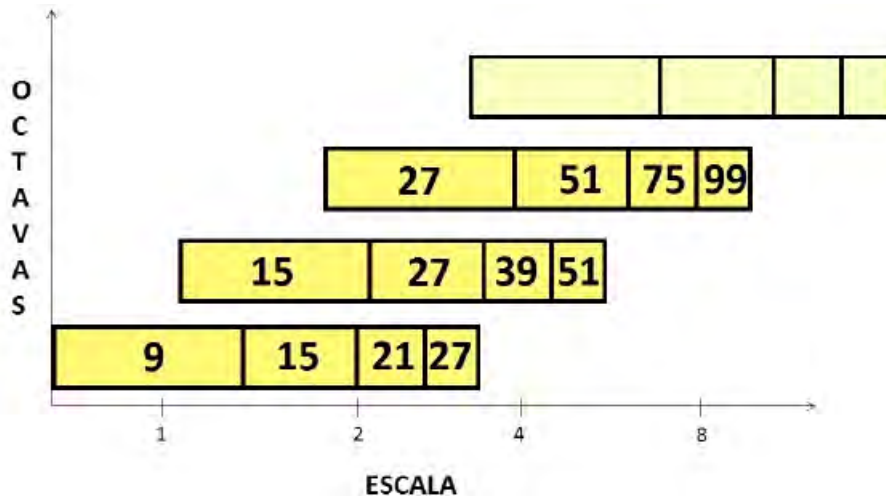
$$H(p, \sigma) = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix} \quad (1)$$

Donde:

$$L_{xx}(p, x) = \frac{\vartheta^2}{\vartheta x^2} \cdot g(\sigma) \quad (2)$$

De igual manera se define  $L_{xy}(p, \sigma)$ ,  $L_{yx}(p, \sigma)$ ,  $L_{yy}(p, \sigma)$ .

El método SURF tiene ciertas limitaciones con los filtros gaussianos, por lo que utiliza un filtro de caja, el cual consiste en hacer una estimación de las derivadas parciales de segundo orden de las gaussianas involucradas.



**Figura 13:** Estructura y tamaño de los filtros utilizados en SURF a diferentes escalas.

Se definen  $D_{xx}$ ,  $D_{yy}$ ,  $D_{xy}$  como las aproximaciones de las derivadas parciales. Se define el determinante de la matriz Hessiana como:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0,9D_{xy})^2 \quad (3)$$

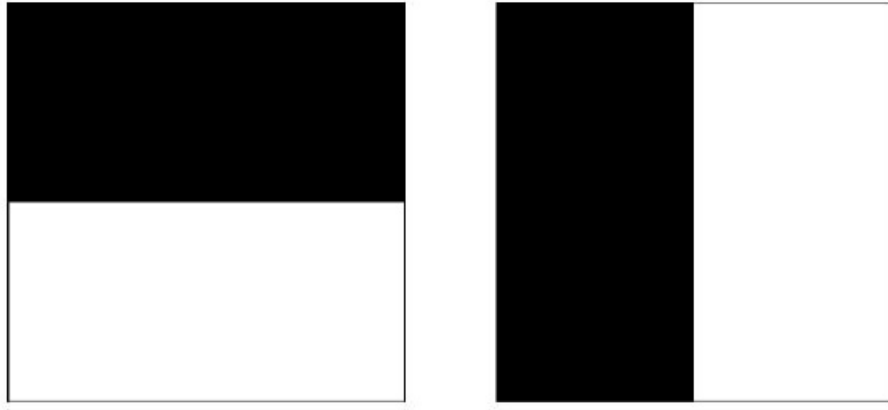
Con el filtro de caja no se es necesario aplicar ningun filtro más a la salida. De primera se aplica el filtro de caja con una dimension de 9x9, obteniendo una salida con una escala inicial de  $a=1.2$  que se corresponde a una gaussiana de  $\sigma=1.2$ .

El algoritmo SURF el espacio-escala esta dividido en octavas. Las escalas tienen un número fijo de imágenes que es el resultado de la convolución de la misma imagen original con una serie de filtros cada vez mas grandes.

Para poder calcular la posición de todos los keypoints en todas las posibles escalas se eliminan todos aquellos puntos que no cumplan con la condición de un vecindario máximo 3x3x3. Por esta razón, el máximo determinante de la matriz Hessiana es interpolado en la escala y posición de la imagen.

## DETERMINACIÓN DE LA ORIENTACION

Cuando ya se ha obtenido los buenos keypoints, el siguiente paso es la determinación de la orientación, haciéndolos invariantes ante cambios en la rotación.



**Figura 14:** Función de Harr para el método SURF

Se calcula la respuesta de Harr con una dirección en  $x$  e  $y$ , como aparece en la figura anterior, en un entorno circular de radio  $6s$  (donde  $s$  es la escala del punto centrado) en el keypoint. Las respuestas onduladas varían en función de  $s$ , cuanto mayor sea la escala mayor será la dimensión de la respuesta. Seguidamente se usa de nuevo las imágenes integrales consiguiendo un filtrado más rápido.

Las respuestas que son obtenidas se ponderan mediante una gaussiana con  $\sigma = 2,5s$  centrada en el keypoint. La orientación dominante la obtenemos mediante la suma del conjunto de respuestas en una ventana de orientación variable un ángulo espacial. Con este parámetro obtenemos de forma experimental y permite cubrir aproximadamente  $\frac{3}{\pi}$  radianes. Con la suma de las dos respuestas, la vertical y horizontal se construye un nuevo vector, con la orientación determinada por el vector de mayor longitud.

## CONSTRUCCIÓN DEL DESCRIPTOR

Cuando se han conseguido los keypoints se procede a crear un descriptor para el almacenamiento. Para almacenar la información, coge una región cuadrada con la orientación calculada en el paso anterior y centrada en el keypoint de tamaño  $20s$ . Se reducen las regiones a subregiones de tamaño  $4 \times 4$  y para cada subregión se determina las características en puntos diferenciados por unas regiones de tamaño  $5 \times 5$ . Las respuestas de Haar para cada una de las direcciones,  $d_x$  y  $d_y$ , con

referencia a la orientación del punto de interés. Con estas respuestas se ponderar con una gaussiana de  $\sigma = 3,3s$ . Mediante esta acción se consigue robustez contra posibles deformaciones geométricas y errores de posicionamiento.

Con las respuestas de cada subregion,  $d_x$  y  $d_y$ , se suman consiguiendo la información del descriptor. Para obtener la información de la polaridad se suma  $|d_x|$  y  $|d_y|$ .

Las subregiones del vector descriptor del método SURF tiene cuatro conjuntos de información por cada elemento y con una extensión de 64 elementos.

Consultar los artículos [6][12] de la bibliografía.

### 2.3.2. Método SIFT

SIFT[10](Scale-Invariant Feature Transform) hace referencia a un algoritmo de búsqueda de keypoints. Lo desarrolló David Lowe en 1999 en la universidad British Columbia en Vancouver, Canadá. Se trata de un algoritmo robusto ante:

- Cambio en la escala.
- Cambio en el giro.
- Ruido en la imagen.
- Cambio en la iluminación.

Es un detector con un buen rendimiento y precisión en la búsqueda de keypoints. Su tiempo de respuesta es aceptable, aunque no es muy rápido. Dispone de un algoritmo de búsqueda de keypoints y la posibilidad de crear sus propios descriptors. Su punto fuerte es la posibilidad de crear una gran cantidad de descriptors estables y robustos ante cambios.

El método SIFT se divide en cuatro procesos:

- Identificación de max y min.

- Filtro de los puntos de interés.
- Determinación de la orientación.
- Construcción de los descriptors.

En las siguientes paginas describiremos las etapas del algoritmo SIFT.

## IDENTIFICACIÓN DE MAX Y MIN

Esta primera etapa realiza varios filtros encadenados para así sacar los valores extremos de los keypoints.

La primera operación es determinar la posición y escala de forma repetida a diferentes vistas del objeto. El conjunto de estas vistas es filtrado por un filtro gaussiano. Los puntos obtenidos son los mínimos y máximos resultantes de las restas de los filtros ejecutados a escalas diferentes

Se define  $L(x,y,\sigma)$  como el espacio de la escala de la imagen, resultante de convolucionar la imagen  $I(x,y)$  con la Gaussiana de escala variable  $G(x,y,\sigma)$  de la siguiente manera:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4)$$

Donde la Gaussiana viene establecida mediante la expresión matemática:

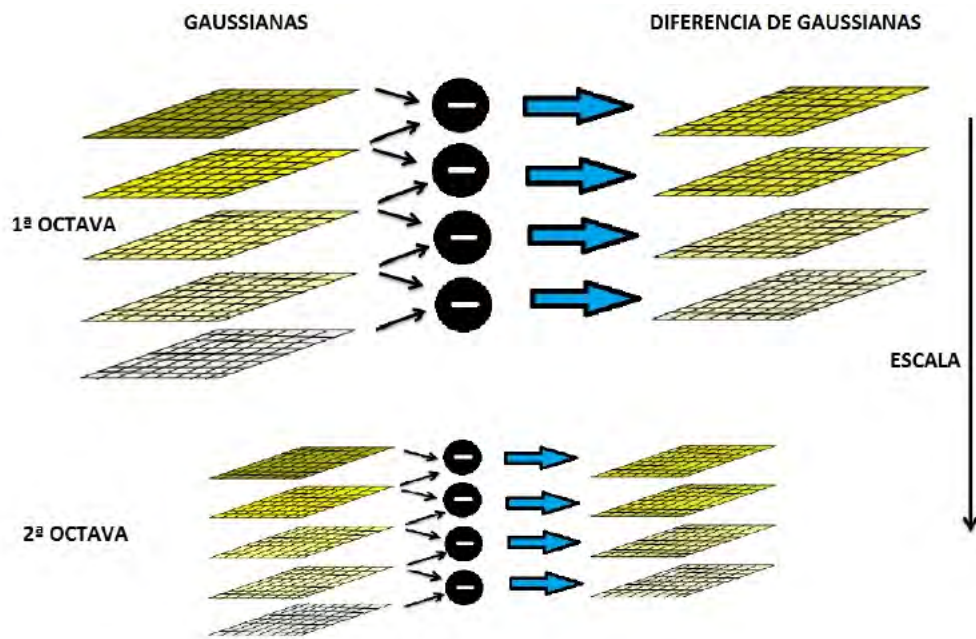
$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}} \quad (5)$$

Para poder obtener un algoritmo que sea más eficaz, que obtenga más keypoints estables se realiza convoluciones basadas en la Diferencia de Gaussianas (DoG) en el espacio-escala:

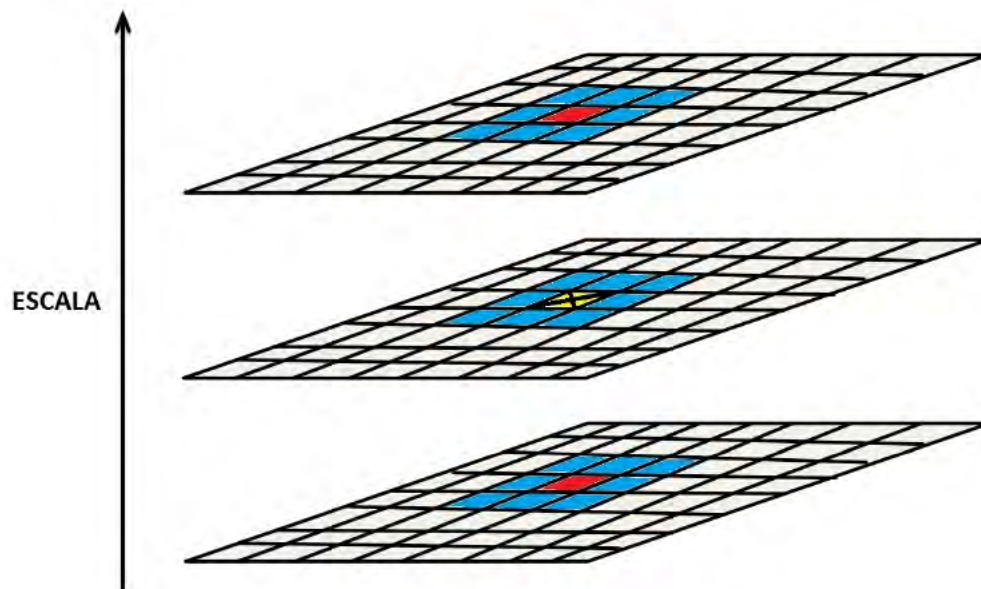
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (6)$$

Donde  $D(x,y,\sigma)$  representa la convolución de la imagen con la diferencia de las

gaussianas en  $\sigma$  y  $k\sigma$ , con  $k$  siendo factor de separación entre escalas definido por  $k = 2^{\frac{1}{s}}$ . Con esta fórmula se puede apreciar que al final  $D(x,y,\sigma)$ , la cual es la DoG, se consigue mediante la resta de dos imágenes con diferentes escalas, así ahorrando tiempo de procesamiento.



**Figura 15:** Diferencia Gaussianas en la escala-espacio



David Lowe en su artículo, unos valores óptimos para la aplicación de este paso serían: número de octavas = 4, número de niveles de escala = 5 y  $\sigma = 1,6$ .

Los puntos candidatos son mínimos y máximos locales de la DoG a las escalas utilizadas. La siguiente operación es la comparación de cada pixel con sus vecinos adyacentes y también con los nueve vecinos de las escalas superior e inferior. Se selecciona como posible keypoint si el pixel analizado es un máximo o un mínimo.

**FILTRO DE LOS KEYPOINTS** Se eliminan aquellos que han pasado por los filtros anteriores pero que no son adecuados por su localización, contraste, etc. Además se establecen varios condicionales:

-Para eliminar los puntos con un bajo contraste hay que obtener la ecuación de Taylor de segundo orden de  $D(x,y,\sigma)$ , obteniendo:

$$D(\hat{p}) = D + \frac{1}{2} \frac{\delta D^T}{\delta p} \hat{p} \quad (7)$$

Si el valor de  $D(\hat{p})$  es mayor a 0.03 el keypoint es descartado.

-Se interpolan cada punto para saber la posición con precisión. Para ello se aplica la expresión de Taylor de la ecuación de DoG que viene dada por:

$$D(p) = D + \frac{\delta D^T}{\delta p} p + \frac{1}{2} p^T \frac{\delta^2 D}{\delta p^2} p \quad (8)$$

En la que p es el keypoint a estudiar. Para poder hacer las comprobaciones se debe derivar respecto a p y particularizar para p=0. Si cualquiera de los valores de la derivada es mayor a 0.5, el keypoint será descartado por el algoritmo.

-Se eliminan los puntos que indiquen un borde puesto que la función DoG tiene una alta respuesta a los bordes.

Si un pico está débilmente definido en la función DoG indica una larga curvatura en la dirección del borde pero pequeña en la perpendicular. Dicha curvatura se evalúa a partir de matriz hessiana de 2x2 en el punto.

$$\begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix} \quad (9)$$

Con  $H$  se puede conseguir la suma de sus autovalores y el producto del determinante:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (10)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (11)$$

Donde  $\alpha$  es el mayor de los autovalores y  $\beta$  es el menor. Se define  $r$  como la relación que hay entre los autovalores  $\alpha = r\beta$ , relacionando así las dos ecuaciones anteriores, obtenemos:

$$\frac{(Tr(H))^2}{det(H)} = \frac{((\alpha + \beta))^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (12)$$

Solo depende de la relación existente entre los autovalores. Para poder realizar la relación entre curvaturas hay que analizar la siguiente ecuación:

$$\frac{(Tr(H))^2}{det(H)} < \frac{(r + 1)^2}{r} \quad (13)$$

Un valor de  $r$  razonable para la relación seria  $r=10$ , si la ecuación anterior no se cumple el keypoint se elimina.

## DETERMINACIÓN DE LA ORIENTACIÓN

En este paso se quiere dar al keypoint de una robustez ante cambios en la rotación. Se saca el histograma del gradiente de la rotación relativo a los puntos vecinos utilizando la imagen gaussiana de escala más proxima al keypoint. Para cada imagen procesada se define una magnitud  $m(x,y)$  y una rotación  $\theta(x,y)$ :

El valor del gradiente se pondera a cada vecino y se asocia una gaussiana de  $1.5\sigma$  con respecto al keypoint. Los picos del histograma son las orientaciones dominantes.

## CONSTRUCCIÓN DE LOS DESCRIPTORES

Este paso se tiene una lista de keypoints, y cada uno de ellos obtienen la información de orientación, escala y posición. El último paso es la creación de los descriptors para almacenar toda la información de los puntos.



Para la creación de los descriptors el algoritmo usa los histogramas de orientación. Los descriptors deben permanecer invariantes ante cambios en la imagen y poder ser distinguidos en un par de imágenes.

Para calcular el descriptor se usa el histograma de orientación en una región de dimension 4x4 en el entorno del punto. Estos histogramas se van construyendo en base a la orientación del punto principal y se pondera por la magnitud del gradiente asociado y por una gaussiana  $1,5\sigma$  del punto estudiado.

Las dimensiones del descriptor componen un vector de 128 elementos. Para darle robustez contra cambios de luminosidad se le aplica una última normalización.

Consultar los articulos [10][12][3] de la bibliografía.

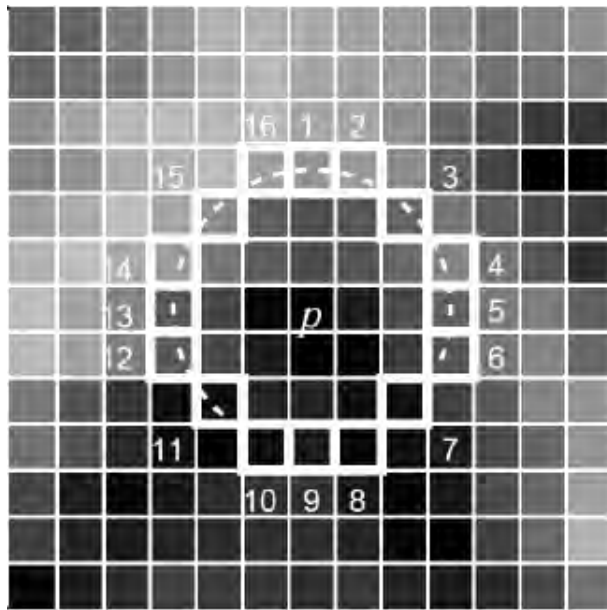
### 2.3.3. Método FAST

FAST (Features from Accelerated Segment Test) es otro algoritmo de búsqueda de keypoints. Fue desarrollado por Edward Rosten y Tom Drummond[13] en 2006. Se trata de un método de detección de esquinas con un bajo coste computacional, por esto lo hace ideal para el uso en computación visual en la robótica.

Con diferencia de los algoritmos de SURF y SIFT, este algoritmo no tiene el poder de crear los descriptors, por lo que se debe usar los descriptors de uno de los algoritmos anteriormente descritos.

### DETECTOR DE PRUEBAS SEGMENTO

Este método usa un círculo de píxeles de Bresenham con un radio 3 para la clasificar si el punto central es una esquina. Se le asigna a cada uno de los píxeles del círculo un número del 1 al 16.



**Figura 16:** Ciruclo Braseham de circulo 3

Se fija la intensidad del pixel candidato  $I_p$ . El brillo es analizado del resto de los puntos del circulo y se compara con  $I_p$ . Si se cumple una de las siguientes condiciones el punto candidato sera una esquina:

- Un conjunto de N pixeles contiguos S, para todo x perteneciente a S, la intensidad del punto x:  $I_x < I_p - t$ .
- Un conjunto de N pixeles contiguos S, para todo x perteneciente a S, la intensidad del punto x:  $I_x > I_p + t$ .

Estas condiciones se van a denominar a partir de ahora condiciones de umbral. Existe una compensación con la elección de N y t. Este algoritmo al ser muy veloz, no es muy precioso.

**ESTUDIO DE ALTA VELOCIDAD** Para conseguir la velocidad del algoritmo primero se compara la intensidad de los pixeles 1, 5, 9 y 13 del círculo con  $I_p$ . Para que exista un keypoint se debe satisfacer el criterio del umbral al menos en 3 de los 4 puntos. Si se cumple esta condición, se analizan los 12 de los 16 puntos del circulo cumplan con el umbral.

Aunque este algoritmo de ensayo tiene varias debilidades:

-La prueba de alta velocidad no se puede generalizar bien para  $N_{12}$ . Para este caso, puede que sólo 2 de las 4 comprobaciones de brillo de píxeles cumplan condiciones de umbral.

-La eficiencia del detector depende de la elección y el orden de los píxeles de prueba seleccionados. Sin embargo, es poco probable que los píxeles seleccionados tengan las condiciones últimas de distribución de las apariencias de la esquina.

-Muchas características se detectan unas a otras.

## MEJORA DEL ALGORITMO CON EL APRENDIZAJE

Para mejorar los dos primeros puntos débiles se introduce un enfoque de trabajo que está basado en el aprendizaje de la máquina para ayudar a mejorar el algoritmo. Este aprendizaje funciona en dos etapas, la detección de esquinas con un valor  $N$  se procesa en un conjunto de imágenes de entretenimiento. Se aplica el algoritmo de la forma más sencilla y se comparan con unos valores de umbral apropiados.

Para el punto candidato  $p$ , cada localización del punto se denota como  $p \rightarrow x$ . El estado de cada pixel  $S_{p \rightarrow x}$  debería pertenecer a uno de los siguientes estados:

- d,  $I_{p \rightarrow x} \leq I_p - t$  (más oscuro)
- s,  $I_p - t \leq I_{p \rightarrow x} \leq I_p + t$  (igual)
- b,  $I_{p \rightarrow x} \geq I_p + t$  (más claro)

El vector se puede dividir en tres diferentes particiones,  $P_d, P_s$  y  $P_b$  en función de los valores obtenidos:

$$-P_d = \{p \in P : I_{p \rightarrow x} = d\}$$

$$-P_s = \{p \in P : I_{p \rightarrow x} = s\}$$

$$-P_b = \{p \in P : I_{p \rightarrow x} = b\}$$

La segunda etapa se declara una variable booleana  $k_p$  que indica si es o no un keypoint. Se aplica el algoritmo ID3, este es un algoritmo de árbol de decisiones, para consultar cada subconjunto utilizando la variable  $K_p$  para conocer cuál es la clase verdadera.

El método funciona con el principio de minimización de la entropía. Se utiliza la variable  $K_p$  que mide la cantidad de información de  $p$  para ser una esquina. Para un conjunto de píxeles  $Q$ , la entropía total de  $K_Q$  es:

$$-H(Q) = (c+n)\log_2(c+n) - c\log_2(c) - n\log_2(n)$$

Donde  $c$  es el número de esquinas y  $n$  es el número de no esquinas.

Se aplica de forma recursiva la minimización de la entropía a los tres subgrupos. El proceso termina cuando la entropía de un subgrupo es cero, de este modo que todos los píxeles de ese subgrupo son o no esquinas, es decir, keypoints.

Consultar los artículos [13][15] de la bibliografía.

#### 2.3.4. Método AGAST

AGAST (Adaptative and Generic Corner Detection Based on the Accelerated Segment Test) es un método más eficiente que FAST ya que utiliza un espacio de configuración más detallado, por lo que se consigue una solución mas eficiente. Para conseguir esto, se evalúa solamente una pregunta por ciclo. La idea es la siguiente: elegir un pixel para probar y plantear solo una pregunta. La pregunta entonces se evalúa para el pixel escogido y la respuesta que se obtiene se utiliza para decidir el siguiente pixel y la pregunta que se hará. Se busca una esquina, por lo tanto, se reduce a travesar un árbol de decisiones binarias. Ya que, se requiere especificar un pixel a evaluar y el tipo de pregunta que se va a realizar. Por consiguiente, la configuración espacial aumenta por la adición de dos estados más: no más claro y no más oscuro. Un estado de un pixel respecto al núcleo, denotado por  $n \rightarrow x$ , es asignado por:

$$S_{n \rightarrow x} = \begin{cases} d, I_{n \rightarrow x} < I_n - t(\text{oscuro}) \\ \bar{d}, I_{n \rightarrow x} \not< I_n - t \bigwedge S'_{n \rightarrow x} = u(\text{nooscuro}) \\ s, I_{n \rightarrow x} \not< I_n - t \bigwedge S'_{n \rightarrow x} = \bar{b}(\text{similar}) \\ s, I_{n \rightarrow x} \not> I_n + t \bigwedge S'_{n \rightarrow x} = \bar{d}(\text{similar}) \\ \bar{b}, I_{n \rightarrow x} \not> I_n + t \bigwedge S'_{n \rightarrow x} = u(\text{noclaro}) \\ b, I_{n \rightarrow x} > I_n + t(\text{claro}) \end{cases} \quad (14)$$

Donde  $S'_{n \rightarrow x}$  es el estado precedente,  $I$  es la luminosidad de un pixel y  $u$  es el estado que todavía no es conocido. Esto resulta en un árbol binario representativo, permitiendo una sola evaluación en cada nodo. Teniendo en cuenta esto aumenta el espacio de configuración a  $6^N$ , cuyo rendimiento  $6^{16} \approx 2 \cdot 10^{12}$  posibles nodos para  $N = 16$ .

Consultar el artículo [16] de la bibliografía.

### 3. Algoritmo

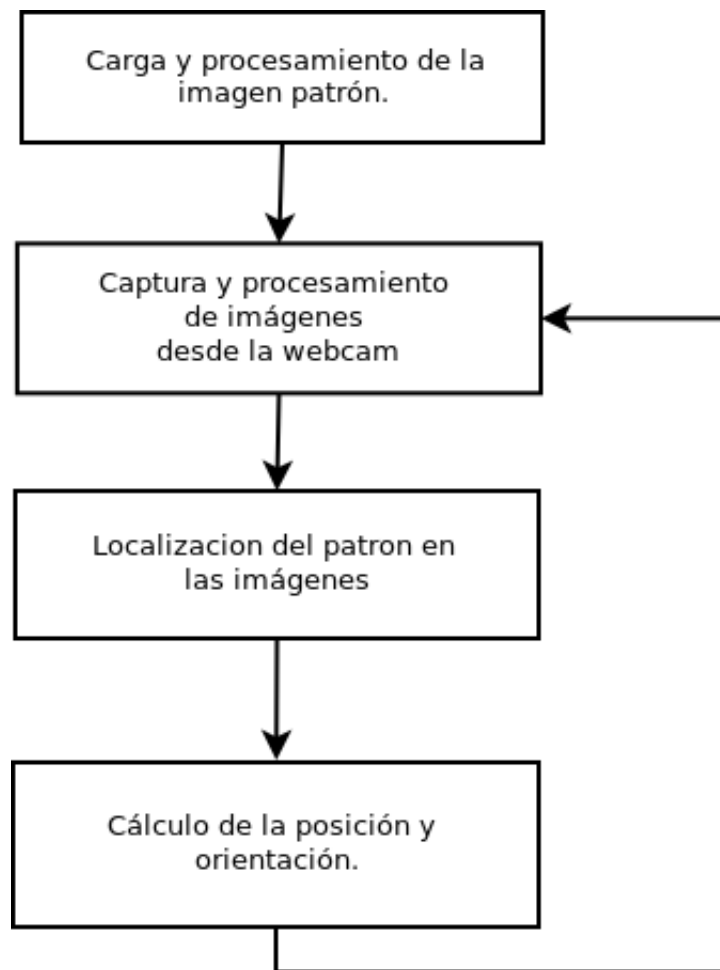
### 3.1. Aspectos generales

En este capítulo se expone la estructura general del código utilizado y analizar las estrategias de diseño. Se expondrán los objetivos y condiciones que debe cumplir el código. Seguidamente, se muestra la estructura que va a tener. Y por último se explica de forma detalla las secciones del código.

#### 3.1.1. Objetivos

El objetivo que tiene que cumplir el código es la extracción de los puntos característicos de una imagen patrón, para poder encontrar esos mismos puntos en las imágenes a tiempo real.

Se puede observar la estructura del algoritmo en el diagrama de flujo propuesto. El código es un método iterativo, es decir, desarrolla su acción dentro de un bucle, que puede ser definido o infinito. Dispone una linea principal de acción, pero dentro de esa linea de acción habra varias líneas de acción secundarias, que seguirá según las condiciones que se cumplan.



**Figura 17:** Diagrama reducido del código

Cuando se ejecuta, lo que se hace es cargar la imagen patrón, es decir, se da valores a una matriz con los correspondientes valores de la imagen y se obtienen los keypoints del patrón. A continuación nos metemos en un bucle que va obteniendo los frame del vídeo que va capturando la cámara y se van obteniendo los keypoints de ese frame.

Cuando ya se tiene los keypoints de la imagen patrón y de la imagen obtenida por la cámara, se comparan y se guardan los puntos que son comunes en las dos imágenes. Con esos puntos guardados se localiza la misma imagen que tiene el patrón en la imagen a tiempo real. Ya localizado se calcula la posición y la orientación del dron respecto al patrón.

Una vez aclarado este punto, se puede definir el funcionamiento básico de cada parte del algoritmo:



- Obtención del patrón. Se carga una imagen almacenada en la memoria.
- Extracción keypoints. se obtiene sus keypoints y se guarda la información de los keypoints en los descriptors.
- Obtención imágenes a comprobar. Se entra en un bucle y se obtiene imágenes desde la cámara.
- Extracción de keypoints. se obtienen sus keypoints y se guardan la información de los keypoints en los descriptors.
- Criba de los keypoints. Se hace una selección de los keypoints para solamente quedarnos con los que nos interesa.
- Localización objeto. Con la selección de keypoints se encuentra el objeto en la imagen que se obtiene de la cámara.
- Triangulación. Mediante este método se obtiene la posición del dron respecto al objeto. Estas coordenadas son  $x, y, \theta$  y  $h$ (altura). -Envío coordenadas. Se envían las coordenadas obtenidas al control del dron.

### 3.1.2. Datos de entrada

El algoritmo hace uso de información del exterior que es necesaria para realizar las operaciones. Las entradas del algoritmo son las siguientes:

- Imagen patrón(imagen 1). Se trata de una imagen en color RGB con un ancho y alto determinados. Esta imagen es obtenida de la memoria, ya que es una imagen que elegimos nosotros.
- Imagen a tiempo real(imagen 2). Se trata de una imagen a color RGB obtenida a tiempo real con un ancho y alto determinado por la resolución de la cámara.
- Matriz de calibración de la cámara. Es una matriz de  $3 \times 3$  que contiene la información de la distorsión de la lente, distancia focal, etc. El dato que vamos a necesitar para poder calcular la altura es la distancia focal además de conocer las medidas del patrón.

### 3.1.3. Datos de Salida

Los datos que nos proporciona el código son las coordenadas del patrón respecto al dron mediante una resta de coordenadas. Esta información se guardara en un vector de puntos para ser enviada a la parte de control.

## 3.2. Algoritmo Teórico

El algoritmo se compone de varias etapas denominadas obtención patrón y sus keypoints, obtencion imágenes a tiempo real y sus keypoints, comparación de keypoints, localización objeto y calculo de posición y altura.

El siguiente cuadro se puede ver el algoritmo de iteración y como se accede a cada una de las etapas:

---

### Algoritmo 1 Obtención de la posición

---

**Entrada:**Imagen 1, Imagen 2.

**Salida:**Coordenadas de posición.

1:**mientras** el programa se ejecute. **hacer**.

2:Obtención y discriminación de los keypoints.

3:Localización objeto.

4:Calculo posición y orientación.

5: **fin mientras**

Este algoritmo representa el funcionamiento general de la estructura, independientemente de si el algoritmo se ejecuta por primera vez o no.

El siguiente algoritmo se va a desarrollar de forma teórica todos los pasos que recorre el algoritmo para su comprensión.

### 3.3. Desarrollo del algoritmo

Dentro de esta apartado se explicara de forma detallada el funcionamiento de cada sección del código, exponiendo fundamentos teóricos.

#### 3.3.1. Obtención y discriminación de los keypoints

Como primer paso del algoritmo, en esta sección se va a introducir las variables de entrada imagen 1 (patrón) e imagen 2, serán computadas para la extracción de los keypoints. El siguiente paso será almacenar en vectores de keypoints y compararlos entre sí para poder eliminar los keypoints que no coinciden en las dos imágenes.

El funcionamiento de este bloque se puede ver en el siguiente algoritmo:

---

**Algoritmo 2 Extracción keypoints y comparación**

---

**Entrada:**Imagen 1, Imagen 2.

**Salida** Keypoints deseados.

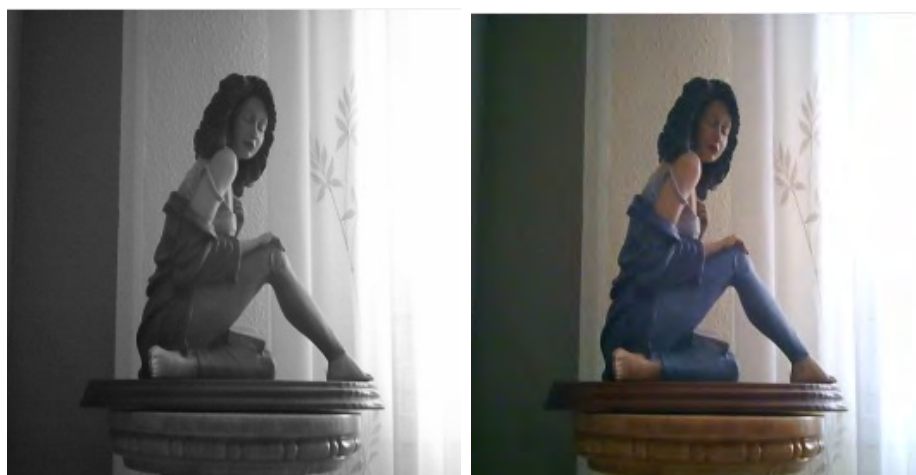
- 1:Obtención de la imagen 1 (patrón) y pasar la imagen 1 de escala RGB a escala de grises.
- 2:Obtención y almacenamiento de los keypoints de la imagen 1.
- 3:Obtención de la imagen 2(imágenes a tiempo real) y pasar la imagen de escala RGB a escala de grises.
- 4:Obtención y almacenamiento de los keypoints de la imagen 2.
- 5:Comparación de los keypoints.
- 6:Eliminación de los keypoints que no son comunes.

#### OBTENCIÓN DE LAS IMÁGENES

La parte básica pero fundamenta del algoritmo es la obtención de las imágenes, ya sea el patrón como las imágenes obtenidas a tiempo real. De ellas depende en gran cantidad que el algoritmo tenga unos resultados bueno o malos. En este caso las dos imágenes tendrán algunas características similares otras no:

- La imagen 1 tendrá una resolución determinada por el usuario.
- La imagen 2 tendrá una resolución definida por la configuración de la cámara, 640x480.
- Las dos imágenes están en la escala de color. La escala más común la RGB.

Se debe tener en cuenta que los extractores de keypoints no pueden obtener los keypoints si la imagen esta en la escala de color. Para conseguir los keypoints hay que pasar la imagen a nivel de grises.



**Figura 18:** Tipos de escalas de imagen. A la izq escala en gris. A la der escala en color

Se establece una escala de 255 tonalidades de gris, que van desde el blanco (0) al negro (255). Cada tonalidad de color de la imagen RGB se le asigna a una tonalidad de gris. Se coge el valor del color RGB de cada pixel y se busca su equivalencia en la escala de grises y se representa su valor en la misma posición que el pixel de color.

## OBTENCIÓN DE LOS KEYPOINTS

Ya con las imágenes en la escala de grises se puede proceder a encontrar los keypoints de cada imagen. Para ellos se necesita crear dos vectores donde se va a guardar la información. Estos vectores tienen el formato de keypoint y una dimension igual al número de ellos que contenga cada imagen. También se crean los descriptors de cada imagen, que es donde se van a almacenar la información que contiene cada keypoints en la imagen.

Se puede usar cualquiera de los algoritmos de extracción de keypoints que hemos

descrito en este documento, los métodos SIFT, SURF, FAST, AGAST. En todos ellos se usa el mismo método de funcionamiento aunque el método SIFT y SURF necesitan texturas, es decir, una imagen de color, para poder obtener los keypoints, en cambio el método AGAST no necesita que la imagen tenga textura.

1. Se crea los vectores de keypoints y los descriptors para la imagen 1.
2. Se extraen los keypoints de la imagen 1 y se guardan en su vector.
3. Se guarda la información de los keypoints en los descriptors.
4. Se crean los vectores de keypoints y los descriptors para la imagen 2.
5. Se extraen los keypoints de la imagen 2 y se guardan en su vector.
6. Se guarda la información de los keypoints en los descriptors.

Los puntos almacenados guardan información como la posición en la imagen, su color, su intensidad, etc. También guarda la información del entorno de cada punto, donde se puede ver cómo interactúa con los puntos vecinos.

## COMPARACIÓN DE LOS KEYPOINTS.

Teniendo ya extraídas las características que definen a las imágenes usadas hay que ver cuáles son compartidas entre ellas. Si dos de los keypoints coinciden en un alto porcentaje tanto en su formación como posición, hay una probabilidad de que estos dos keypoints pertenezcan al mismo punto. También existe una comprobación que aclara más la similitud de los puntos. Si además de compartir la misma información entre los puntos, sus puntos vecinos también son similares en las dos imágenes, se podrá certificar con mayor exactitud que los puntos son los mismos.

Para ello se utiliza unos algoritmos de comparación especializados en esta tarea, como el usado en nuestro algoritmo BruteForce. El funcionamiento de este algoritmo se basa en las siguientes hipótesis:

- 1- Se crean dos vectores, uno para guardar los valores obtenidos en las comparaciones y otro para guardar distancias entre keypoints.
- 2- Para cada keypoints de una imagen 1 se hace una comparación con los de la imagen 2 y se intenta encontrar que haya coincidencia de estos puntos en las imágenes.
- 3- Si se cumple la condición anterior se comprueba los puntos vecinos. Se toman

las dos distancias mínimas a los vecinos y se almacenan en el vector de distancias. De forma paralela, se van almacenando los valores de en el vector de información.

En el momento que se completa este algoritmo ya tenemos suficiente información: información de los keypoints, de la imagen 1 como de la imagen 2, información relativa a la comparación de los keypoints de las dos imágenes y las distancias mínimas a los vecinos con la información relevante de ellos.

El próximo paso del algoritmo es agrupar los puntos con información y la eliminación de los puntos que no aportan información. Con este paso se consigue la optimización del tiempo de ejecución y así poder mejorar la calidad de la localización del objeto. Se optimiza el tiempo, porque a la hora de estudiar la rotación y la translación o de representar los puntos, el algoritmo tiene que estudiar muchos menos puntos que estudiar.

Es necesario hacer una eliminación con la información recibida de la comparación. El proceso que se sigue es el siguiente:

1. Para la criba se evalúan las distancias que se han obtenido. Se hace un cociente entre las dos distancias y se busca que sea menor que un ratio fijado por el programador. Por lo que se puede afinar más la búsqueda de los keypoints que coinciden en las dos imágenes.
2. Si esta comparación es correcta el algoritmo pasa por este punto sin hacer nada. Si no es correcta la comparación, modifica el valor de la comparación y le da un valor negativo. De esta manera lo que se consigue es desechar el valor de la comparación de ese keypoint y así no lo tiene en cuenta en otras operaciones.
3. Cuando ya se ha terminado la comprobación, la información filtrada se pasa a la siguiente etapa del algoritmo.

Consultar los artículos [4][14] de la bibliografía para profundizar más.

### 3.3.2. Homografía y búsqueda de objeto

La siguiente etapa es hacer la homografía. Con la homografía podremos conseguir representar un punto 2D como un vector 3D. Antes de describir el algoritmo vamos a definir el concepto de homografía, además el funcionamiento del método RANSAC de eliminación de puntos exteriores y el método de encontrar el objeto.

#### MATRIZ DE HOMOGRAFÍA

La homografía nos ayudara a representar un punto y localizarlo en un plano, para ello sera suficiente hallar la matriz 3x3 de homografía H. Esta matriz esta considerada una matriz homogénea y solo tiene ocho grados de libertad aunque contenga 9 elementos, es decir, habrá 8 elementos desconocidos para ser resuelto.

La homografía se estima entre imágenes encontrando características correspondientes en esas imágenes. Los algoritmos más típicos hacen uso de puntos característicos correspondientes, aunque otras características pueden ser líneas o esquinas.

Si examinamos cuidadosamente la ecuación podemos realizar una situación especial para nuestro objetivo. Si el objeto que observamos es un plano, podemos asumir que el punto de ese plano esta localizado en  $z=0$ , el resultado de la ecuación seria la siguiente:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r1 & r2 & r3 & t1 \\ r4 & r5 & r6 & t2 \\ r7 & r8 & r9 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

**Figura 19:** Matriz intrínseca con  $z=0$

El  $z=0$  en el punto de la escena hace cancelar la tercera columna de la matriz proyectiva, lo que hace otra vez una matriz de 3x3. Esta matriz especial es la denominada homografía, eso implica circunstancias especiales (aquí, una rotación pura o un objeto plano), un punto del mundo puede ser relacionado a su imagen

por una relación lineal. La relación de la homografía es de la siguiente forma:

$$s \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (15)$$

Donde H es la matriz de 3x3. la relación mantiene un factor de escala representado por s valor de escala.

**RANSAC** El método RANSAC (Random Sample Consensus) es un método robusto más común de estimación para homografías. La idea del algoritmo es bastante simple, para un número de iteraciones, un ejemplo aleatorio de cuatro correspondencias son seleccionadas y una homografía H se computa con las cuatro correspondencias. Cada otra correspondencia son clasificadas como inliers o outliers, dependiendo de su concurrencia con H.

Una cuestión importante cuando aplicamos el algoritmo RANSAC descrito arriba es como decidir como clasificar las correspondencias como inliers o outliers. El objetivo es asignar una distancia mínima t( entre x' y Hx por ejemplo), tal que con una probabilidad  $\alpha$  el punto es un inlier.

Otra cuestión es como decidir cuantas iteraciones hacer en el algoritmo.

$$N = \log(1 - p) / \log(1 - (1 - \epsilon)^s) \quad (16)$$

Donde p es la probabilidad que al menos una de las muestras aleatorias este libre de outliers, s es el número de correspondencia usada en en cada iteración y  $\epsilon$  es la probabilidad que una simple correspondencia sea un outlier. Si  $\epsilon$  es desconocido, el dato puede ser investigado para determinar adaptativamente  $\epsilon$  y N.

## LOCALIZACIÓN DEL OBJETO Y DIBUJADO DEL RECTÁNGULO

Para la localizar el objeto en la imagen a tiempo real, se necesita analizar la matriz de homografía. Para ello se usa un algoritmo que asigna los puntos de la primera



imagen en la segunda, con la información de los keypoints de la imagen 1 y de la imagen 2 y también con la matriz de homografía.

Cuando ya se han pasado los puntos de la imagen 1 a la imagen 2, el siguiente paso es dibujar un rectángulo sobre el objeto para así poder saber que la homografía se ha hecho correctamente y poder calcular su posición.

Para dibujar el rectángulo necesitaremos un vector de puntos para almacenar los puntos que nos proporciona la localización del objeto.

Cuando se tienen los 4 puntos se dibujan líneas entre los puntos para así conseguir el rectángulo deseado.

## DEFINICIÓN DEL ALGORITMO

El algoritmo 3 muestra las etapas de la homografía descrito anteriormente.

---

### Algoritmo 3 Homografía y localización del objeto

---

**Entrada:** Keypoints filtrados de la imagen 1 y imagen 2

**Salida** Rectángulo sobre el objeto.

1. **para** 0 hasta N, donde N se determina de forma adaptativa dentro del algoritmo.

**hacer**

2. Selección de 4 correspondencias y computar la matriz H.

3. Computa la homografía.

4. Localización del objeto.

5. Dibujado de rectángulo.

Consultar los artículos[] de la bibliografía para más información.

### 3.3.3. Coordenadas y altura respecto al objeto

En esta etapa vamos a explicar la extracción de las coordenadas y de la altura respecto al objeto localizado antes de definir el algoritmo de esta etapa.

## EXTRACCIÓN DE COORDENADAS

Para la extracción de coordenadas nos haremos uso de ecuaciones matemáticas.

Teniendo en cuenta que disponemos de los puntos que componen el rectángulo, podemos hallar el centro del rectángulo uniendo con recta los puntos paralelos que hacen la mitad de cada lado del rectángulo. La intersección entre las dos rectas sera el punto medio del rectángulo

La diferencia que obtenemos la tenemos en pixeles para conseguir la diferencia en metros, simplemente hay que saber lo que mide en metros un pixel de la cámara.

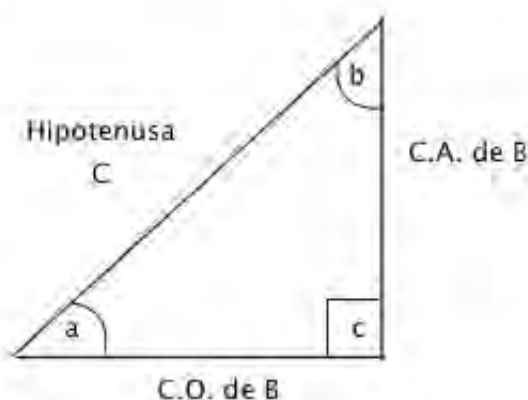
Para pasar los pixeles a mm hay que hacer la siguiente operación:

$$pixelmm = \frac{dimensionensormm}{pixelmaxancho} \quad (17)$$

## EXTRACCIÓN DE LA ORIENTACIÓN

Para obtener la orientación del dron respecto al objeto nos haremos ayuda de las rectas que hemos dibujado para hallar el centro. Con la recta que une los lados estrechos como referencia del ángulo cero del objeto. Desde el centro de la imagen dibujaremos otra recta que sera el cero del dron.

Ya teniendo las dos rectas como referencias a través de las razones trigonométricas podremos sacar la orientación.



**Figura 20:** Trigonometría

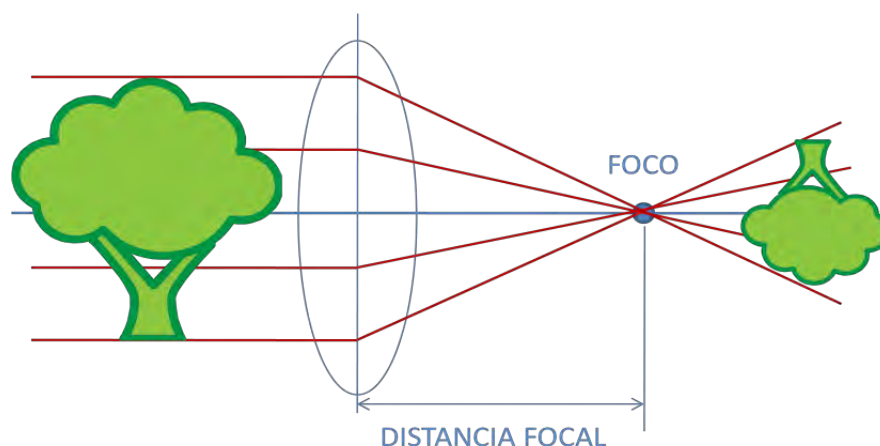
Donde la recta de referencia del patrón es la la hipotenusa y la recta de refe-

rencia de dron seria el cateto contiguo y con la relación siguiente podremos sacar el ángulo:

$$b = \cos^{-1}\left(\frac{cc}{h}\right) \quad (18)$$

## OBTENCIÓN DE LA ALTURA

Para calcular la altura hay que explicar el modelo de cámara pin-hole para entender el proceso. Cuando se capta una imagen se obtienen unas coordenadas 2D del objeto y por lo tanto la profundidad a la que esta el objeto se puede calcular. Para ello necesitaremos de cierta información que sera la distancia focal, nos la proporcionara cuando se calibre la cámara. Esta distancia focal es la mínima distancia a la que estará el objeto.



**Figura 21:** Distancia focal

Para calcular la altura simplemente haciendo una operación matemática se consigue la altura a la que se esta. Para ello necesitaremos saber las dimensiones del objeto.

$$Altura = \frac{dimensionreal * distanciafocal}{dimensionimagen} \quad (19)$$

## DEFINICIÓN DEL ALGORITMO

El algoritmo 4 nos muestra las etapas para la obtención de las coordenadas y la orientación del dron respecto al objeto.

---

### Algoritmo 4 Coordenadas y orientación

---

**Entrada:** Puntos que componen el rectángulo.

**Salida:** Coordenadas y orientación.

1. Búsqueda centro del rectángulo.
2. Se calcula la diferencia entre el centro de la imagen y el centro del rectángulo y también se calcula la altura.
3. Prolongación de las líneas del rectángulo y dibujar línea vertical en el centro de la imagen.
4. Cálculo de la orientación.

## 4. Experimentos y resultados

## 4.1. Introducción

En este capítulo abordaremos las elecciones y demostraciones que se han ido desarrollando a lo largo del documento y también comentaremos los dispositivos usados en los experimentos. Se compondrá de dos bloques. El primer bloque denominado plataforma se definirán los elementos usados para los experimentos. El segundo bloque se mostraran los experimentos y resultados obtenidos.

## 4.2. Plataforma

En este bloque comentaremos los elementos que hemos usado para hacer las pruebas. Lo podemos separar en hardware y software.

### 4.2.1. Hardware

Para poder crear y testear el algoritmo empleado se ha usado un portátil Acer, modelo v3-571G con una ampliación de ram a 8 GB en total para tener suficiente memoria para la ejecución del algoritmo. Para la adquisición del imágenes en tiempo real se ha usado una cámara deportiva, modelo SJ4000+wifi ya que dispone de una buena calidad de imagen y sobre todo al tener un peso reducido no penalizara en el peso del dron.

El dron es un modelo que se ha echo en la universidad, el cuerpo esta hecho mediante una impresora 3D y el resto de sus componentes motores, baterias, etc, no es imprescindible hablar sobre ellos ya que no son necesarios para el algoritmo.

### 4.2.2. Software

De software lo que se ha usado es el sistema operativo Ubuntu una compilación sobre la base Linux, para la creación del algoritmo se ha usado el programa Qtcreator con el añadido del paquete ROS. El paquete ROS contiene tanto las librerías de opencv y las herramientas para la creación de programas de robótica.

### 4.3. Experimentos

En este bloque se mostrarán los experimentos realizados con distintos patrones con muchas texturas como patrones con pocas texturas. Se definirá el método elegido para la extracción de keypoints.

Vamos a explicar que ventajas tiene cada métodos:

-Método SURF. Robusto ante cambios en la imagen. Tiempo de procesamiento reducido.

-Metodo SIFT. Buen rendimiento, precisión y tiempo de ejecución, aunque este es algo mayor que el del método SUFT. Este método es más robusto ante cambios de rotación, escala, iluminación,y aparición de ruido.

-Metodo FAST. Bastante rápido en tiempo de ejecución, buen detector de esquinas. Pero no dispone de control de orientación y necesita de otro método para la creación de descriptors.

-Método AGAST. Detector de esquina, rápida ejecución, mas que el método FAST. Pero necesita de otro método para la creación de los descriptos.

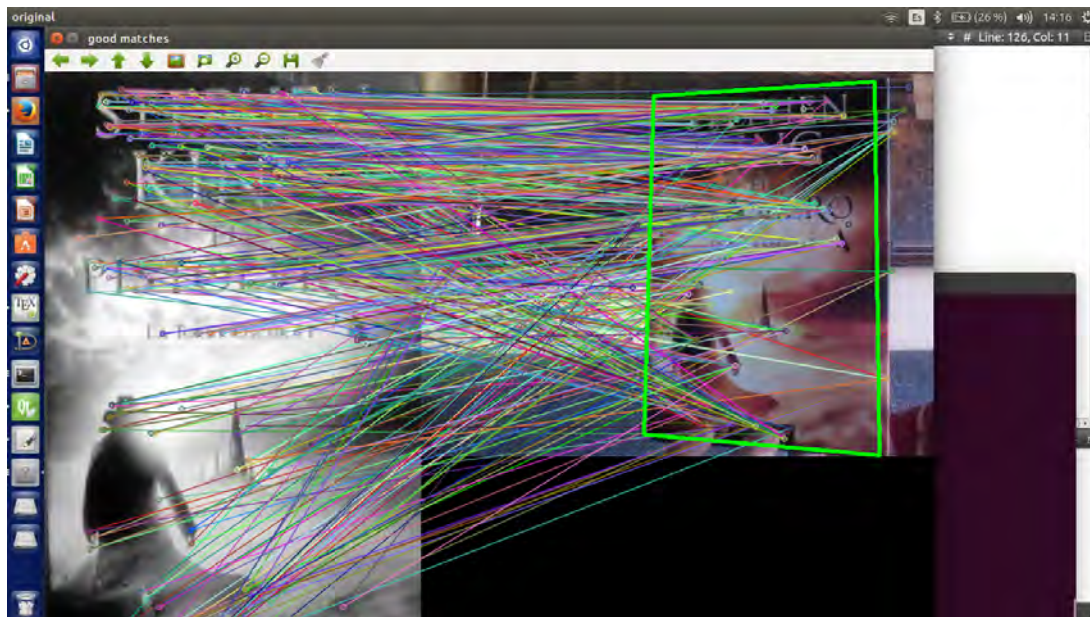
### EXTRACCIÓN DE KEYPOINTS

Se va a mostrar lo números de keypoints que se obtienen con cada método y en cada imagen. En los ejemplos con muchas texturas se ha usado el método SURF descrito anteriormente en este documento y en los ejemplo de poca textura se ha usado el método AGAST para la detección de los keypoints y el método SURF para los descriptors, ya que le método SURF, SIFT necesitan de texturas para trabajar y es el inconveniente con patrones de poca textura que no contienen texturas.

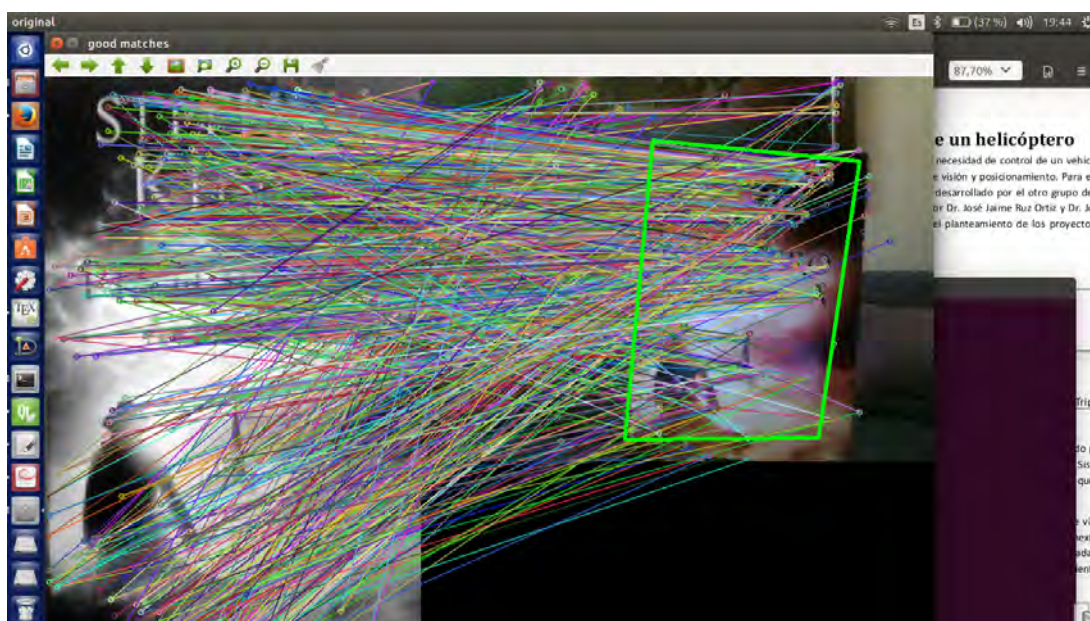
Para la imagen 1 se ha usado la portada de un libro. Como se puede comprobar esta en color y dispone de texturas, por lo que encontrar los keypoints son más fáciles.

método	nº de keypoints
surf	169
sift	757

Por lo que se puede ver se consiguen menos puntos con el método SURF pero en visto en la practica es más rápido que el método SIFT.



**Figura 22:** Experimento imagen 1 con surf



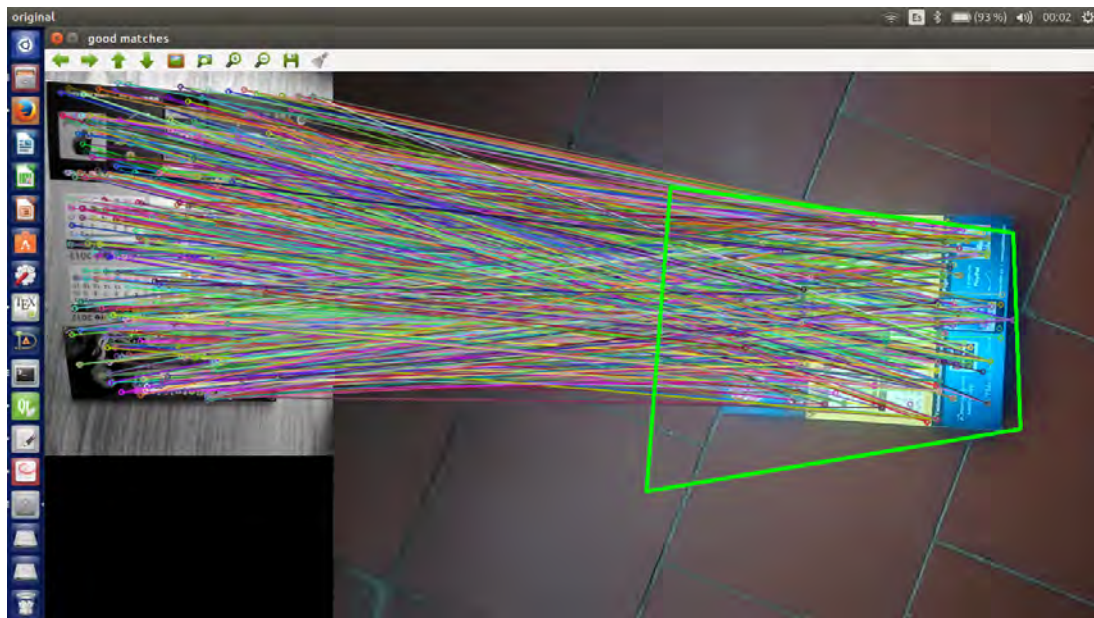
**Figura 23:** Experimento imagen 1 con sift

Para la imagen dos se ha usado un calendario que contiene ilustraciones más pequeñas. Como en el anterior ejemplo el calendario también esta en color.

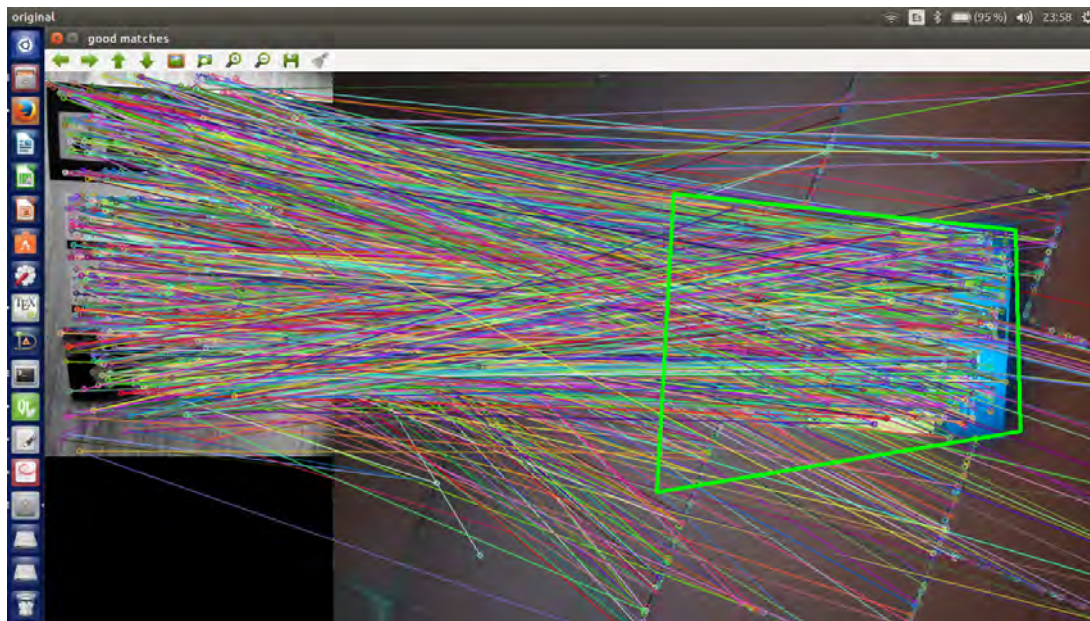


método	nº de keypoints
surf	198
sift	1805

Por lo que se puede comprobar la diferencia de keypoints es muy grande pero con los dos sistemas el resultado es el mismo, la localización del rectángulo. Con el método SURF los puntos están mas concentrados que con el método SIFT. También se puede comprobar que el recuadro que encierra el calendario no se ajusta en la totalidad ,ya que cuando se hizo la captura del patrón el calendario estaba inclinado, por lo que cuando calcula la matriz homografica hay un pequeño error.



**Figura 24:** Experimento imagen 2 con método SURf



**Figura 25:** Experimento imagen 2 con método SIFT

El siguiente experimento se hace con una imagen en de poca textura, ya un patrón normalizado para el aterrizaje. En este caso se hizo dos pruebas: el primero sin mascara y la segunda con mascara.

La primera prueba el inconveniente que tiene es que el patrón al tener pocas texturas cuando haya sombra seleccionara ese punto como bueno y no sera capaz de crear la matriz de homografía correctamente, por lo que en la segunda prueba se probó a poner una mascara para delimitar la zona de búsqueda de los keypoints. Las imágenes y tabla que se muestran son de la segunda prueba.

método	nº de keypoints
agast	110

Lo que se puede comprobar es que el enlace entre los keypoints lo hace correctamente pero no es capaz de crear el rectángulo, por lo que la matriz de homografía no se esta creando correctamente.

Se hace una prueba con el método SURF pero se consiguen la cantidad de 726 keypoints pero con el mismo resultado que con método AGAST. También se hace una prueba con el método SIFT pero este método se rechaza muy rápido ya que

obtiene tantos keypoints que el sistema se vuelve muy lento.

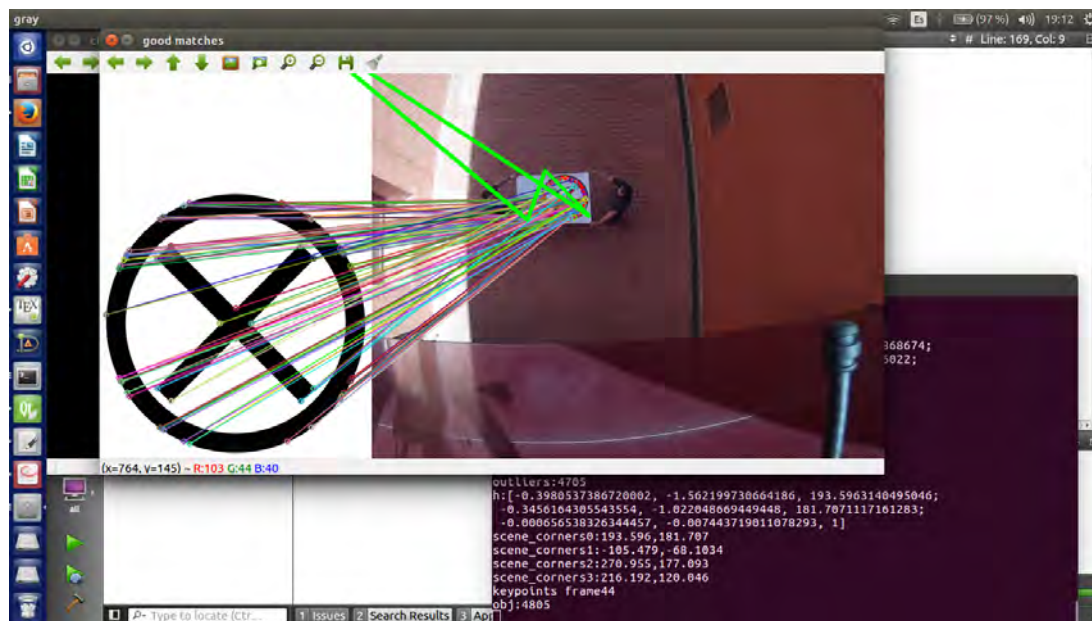


Figura 26: Experimento imagen 3 con método AGAST

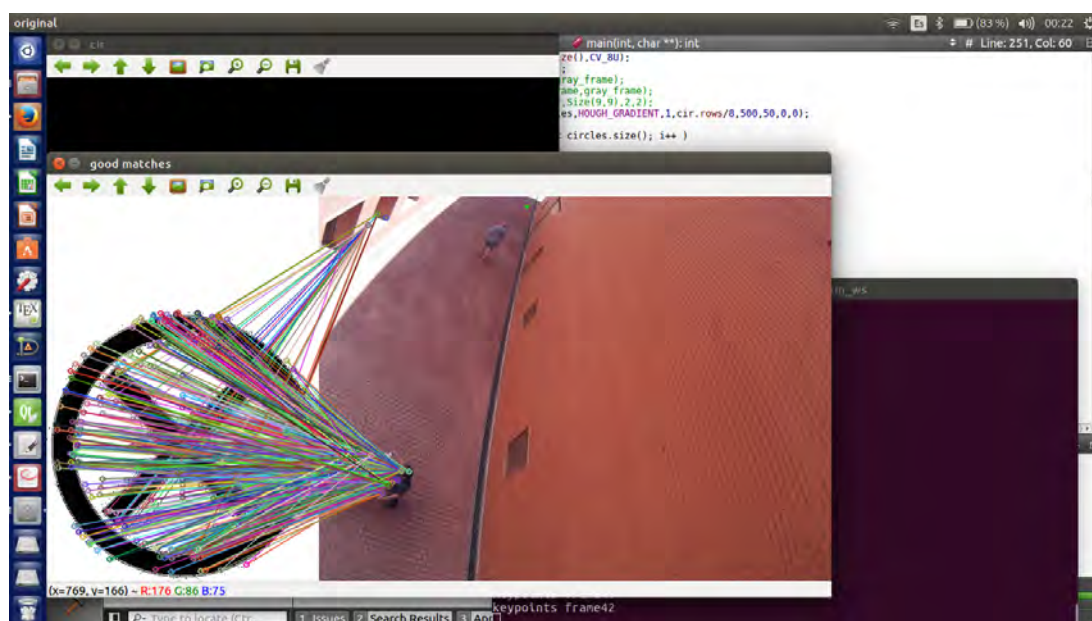


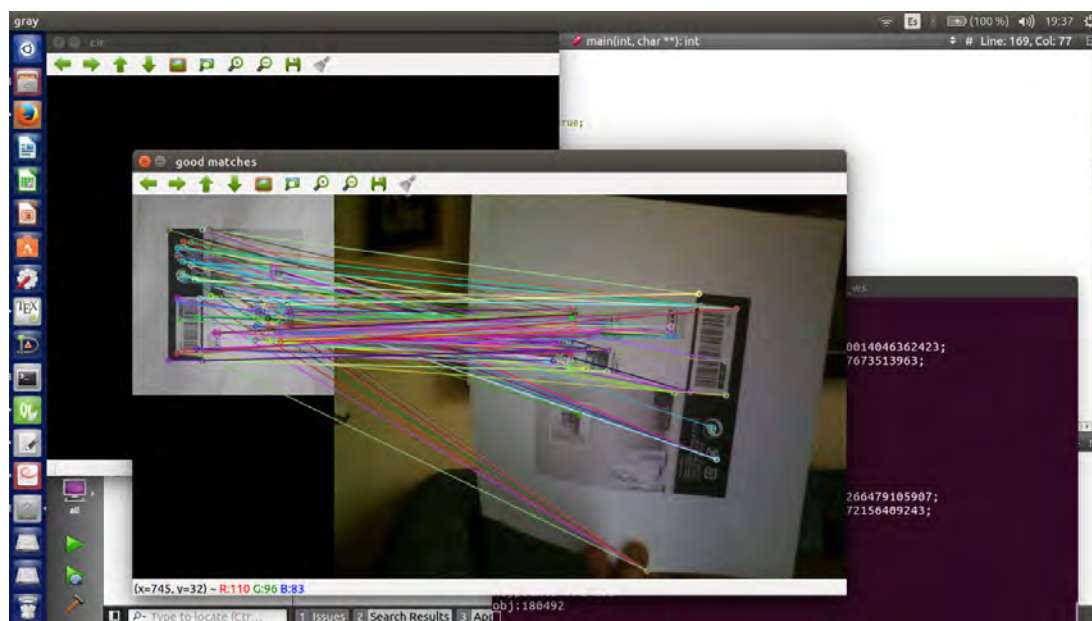
Figura 27: Experimento imagen 3 con método SURF

Se hace un ultimo experimento con otra imagen con poca textura para comprobar si es capaz de reconstruir el recuadro.



método	nº de keypoints
agast	474

El resultado que se obtiene es idéntico al anterior experimento detecta bien y dibuja bien los keypoints coincidentes pero sin el resultado esperado.



**Figura 28:** Experimento imagen 4 con método AGAST

## 5. Conclusiones y futuros trabajos

## 5.1. Introducción

El proyecto abarca la detección de un objeto en imágenes a tiempo real partiendo de un patrón. Como se ha descrito en los apartados anteriores hay diversas alternativas para conseguir la detección del objeto. De todos los métodos descritos se ha elegido el método AGAST ya que ha sido el que mejor ha funcionado para imágenes con poca textura. Para la obtención del resultado se elabora un algoritmo repetitivo para la extracción de la información del entorno, su tratamiento y su transformación. El objetivo es evaluar las prestaciones de este algoritmo.

## 5.2. Contribuciones del proyecto

Las principales aportaciones que se han realizado son:

- Se hace un estudio del arte de las distintas formas de extracción de los puntos característicos de la imagen. Se elabora una explicación de los métodos de extracción de keypoints y descriptors SURF, SIFT, FAST y AGAST, se analiza el funcionamiento, ventajas e inconvenientes de cada uno.
- Se explica el funcionamiento de cada etapa del algoritmo y se muestra el estudio del funcionamiento de cada principio participante y su uso dentro del algoritmo.
- Se desarrolla un algoritmo con el lenguaje C que muestra el resultado. Se razona la elección de los métodos usados para la extracción de keypoints, filtrado y procesado, como la localización y enmarcado del objeto a encontrar.

## 5.3. Conclusión

Este estudio plantea la localización y la obtención de las coordenadas y orientación del dron respecto al objeto a encontrar mediante el tratamiento de las imágenes a tiempo real. Por lo que el proyecto ofrece los siguientes resultados.

- Conseguir los keypoints tiene mucho peso debido a que la obtención de la matriz de homografía viene determinada por la adquisición de los keypoints. Mientras

se hacia el proyecto se han propuesto unos determinados métodos para la adquisición de los keypoints. Para las imágenes con poca textura se opta por el método AGAST para la adquisición de los keypoints y el método SURF para los descriptores.

- Con lo que se consigue con el filtrado de los keypoints es aumentar la mejora de obtener una matriz de homografía más precisa. Seleccionar el método de comparación como BruteForce y los métodos de exclusión de los extremos como es el método RANSAC adquiere mucha importancia y su buena utilización para conseguir una calidad buena del producto.

-Respecto a dibujar el rectángulo sobre el objeto, cuando la imagen tiene poca textura como por ejemplo el patrón de aterrizaje estandarizado no es capaz de encuadrarlo, aunque lo intenta no lo hace bien. Esto es porque la matriz de homografía no es cien por cien correcta y el motivo es al calcular los inliers no están bien localizados por lo que la matriz de homografía no se configura bien.

Por lo que podemos decir que este método de localización de objetos con imágenes con poca textura no es apto para encontrar el objeto cien por cien.

## 5.4. Trabajos futuros

Mientras se realizaba el proyecto han ido surgiendo ideas de mejoras que no se han aplicado al proyecto por falta de tiempo en la investigación. A continuación se exponen las dos ideas que tendrían el mejor resultado.

-También se profundizara más internamente en las matriz de homografia para que el sistema usado en este proyecto pueda funcionar.

-En vez de usar un método de extracción de de puntos característicos, se podría usar un en este proyecto al tener un patrón con un circulo impreso, un detector de círculos. Como es el algoritmo HoughCircles. Este algoritmo se encarga de buscar círculos en la imagen desde un radio min a un radio max que es configurado por el usuario.

-Otro método es el Haar Cascade. Este método es un algoritmo de detección de

objetos basado en clasificadores en cascada. Este método es eficaz lo único que este algoritmo hay que entrenarlo.



## 6. Apéndice

## 6.1. Apéndice A

### PRESUPUESTO DE PROYECTO

En apéndice se muestra justificadamente los costes totales para la realización de este Trabajo Fin de Grado. Los costes se separan en gastos de personal y gastos de material, que se separan en dos tablas.

Primera fase	Buscar documentación	100 horas
Segunda fase	Crear algoritmo	400 horas
Tercera fase	Desarrollo memoria	180 horas

En esta tabla se enseña las tres fases y el tiempo aproximado que ha llevado el proyecto. Por lo que se puede concluir que el tiempo total dedicado ha sido de 680 horas, en cada fase ya están incluidas las horas que se han estado con el tutor. El precio de la mano de obra se propone a 22,5 €/hora netos, por lo que el importe total asciende a 15.300€

Cámara USB	99 €
Ordenador	599 €
Dron	150 €
Software Qtcreator	Gratuito
Librería ROS	Gratuito
Latex editor de texto	Gratuito

En la tabla anterior se adjunta los costes de los materiales tanto hardware como el software utilizados que ascienden a un total de 848 €.

Concepto	Importe €
Coste personal	15.300 €
Coste material	848 €
Total	16.148

## 6.2. Apéndice B

Marco-Legal Para el uso de drones hay unas cierta leyes que hay que cumplir. Tanto para uso comerciales o profesionales como uso particular. Las leyes para uso comercial o profesional son las siguientes:

- El RD Ley 8/2014, de 4 de julio de aprobación de medidas urgentes para el crecimiento, la competitividad y la eficiencia.
- La Ley 18/2014 de 15 de octubre, de aprobación de medidas urgentes para el crecimiento, la competitividad y la eficiencia. Para uso particular se debe cumplir las siguientes leyes:
  - Deben respetar la legislación aeronáutica general, y la regulación que cada Comunidad Autónoma o Municipio establece sobre esta práctica lúdica o deportiva.
  - Ley 18/2014, no precisando de habilitación especial por la AESA.
  - Sujeto a sanciones previsto en la Ley 214/2003.

## BIBLIOGRAFÍA

- [1] <http://www.agmabogados.com/es/regulacion-legal-los-drones-espana/> última acceso 22/09/2017
- [2] Dictionary of military and associated terms. Department os Defense U.S. ,31/10/2009
- [3] [https://en.wikipedia.org/wiki/scale-invariant\\_feature\\_transform](https://en.wikipedia.org/wiki/scale-invariant_feature_transform). Última revisión 31/05/2017.
- [4] D. L. Baggio. Mastering OpenCV with practical computer vision projects, volume Chapter 4: Exploring Structure from Motion Using OpenCV. Packt Publishing Ltd, 2012.
- [5] A. Barrientos, J. del Cerro, P. Gutiérrez, R. San Martín, A. Martínez, and C. Rossi. Vehículos aéreos no tripulados para uso civil. tecnología y aplicaciones. Universidad politécnica de Madrid, Madrid, 2007.
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In Computer vision-ECCV 2006, pages 404-417. Springer, 2006.
- [7] R. R. Delgado. Los drones y sus aplicaciones a la ingeniería civil. Capítulo 14. Aplicaciones al mantenimiento de líneas eléctricas:175-184, 2015.
- [8] M. I. García. Algoritmos de visión para la estimación robusta de pose 3d. pages 67-82, Octubre 2010.

- [9] Y. Huang, S. J. Thomson, W. C. Hoffmann, Y. Lan, and B. K. Fritz. Development and prospect of unmanned aerial vehicle technologies for agricultural production management. *International Journal of Agricultural and Biological Engineering*, 6(3):1-10, 2013.
- [10] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150-1157. Ieee, 1999.
- [11] J. M. Peña-Barragán, F. López-Granados, L. García-Torres, M. Jurado-Expósito, M. S. De La Orden, and A. García-Ferrer. Discriminating cropping systems and agroenvironmental measures by remote sensing. *Agronomy for Sustainable Development*, 28(2):355-362, 2008.
- [12] C. C. Rejado. Los drones y sus aplicaciones a la ingeniería civil. Capítulo 1. Origen y desarrollo de los Sistemas de Aeronaves Pilotadas por Control Remoto:15 32, 2015.
- [13] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508-1515. IEEE, 2005.
- [14] Z. Ting. Emgucv image process: Estimating projective relations in images, Septiembre 2015. <http://yy-programer.blogspot.com.es/2013/07/emgucv-image-process-estimating24.html>.
- [15] D. G. Viswanathan. Features from accelerated segment test (fast), 2009.

- [16] Elmar Mair, Gregory D.Hager, Darius Burschka, Michael Suppa and Gerhard Hirzinger. Adaptative and Generic Corner Detection Based on the Accelerated Segment Test.